

SUPER

Anno 2
n° 8 - Settembre 1985

L. 7.500

Spedizione in
abbonamento
postale Gruppo III/70

8

SUPERSINC
la rivista per gli utenti
dei prodotti
SINCLAIR

sinc

Una pubblicazione della

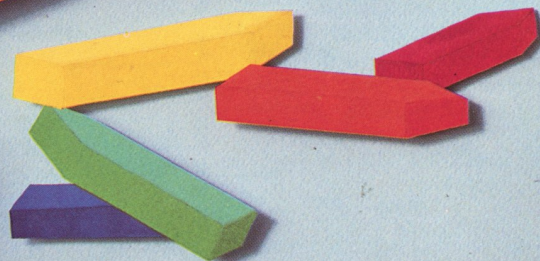
 **J. soft** EDITRICE

in collaborazione con

GRUPPO
EDITORIALE
JACKSON



TEST: SAGA EMPEROR 1
IF... THEN: AVVENTURA
NELL'ADVENTURE - seconda parte
VIDEOCALC PER ZX81
CLASSIFICHE DEI CAMPIONATI
QUARK: FUNZIONI IN 3D
REVERSI
R'BERT



dalla biblioteca Jackson informatica per tutti



Rita Bonelli,
Luciano Pazzucconi,
Fabio Racchi

COMMODORE 16: SEMPRE DI PIÙ

Un libro sul Commodore 16 per approfondire le conoscenze sulla macchina e sul suo BASIC.

cod. 427B Pag. 336
Lire 35.000 Con cassetta

David Lawrence

TECNICHE DI PROGRAMMAZIONE SUL COMMODORE 64

L'arte della buona programmazione alla portata di chiunque possenga un Commodore 64.

cod. 575D Pag. 176
Lire 16.500

Daria Gianni, Carlo Tognoni

MSX: IL BASIC

Il primo libro sul BASIC MSX,

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano
CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esenzione del contributo spese di spedizione:

- Allego assegno della Banca
- Allego fotocopia del versamento su c/c n. 11666203 a voi intestato
- Allego fotocopia di versamento su vaglia postale a voi intestato

n° _____

Nome _____

Cognome _____

Via _____

Cap _____ Città _____ Prov. _____

Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A.

ORDINE
MINIMO
L. 50.000

che unisce le caratteristiche di un manuale di riferimento a quelle di un buon testo didattico di programmazione.

cod. 417D Pag. 216
Lire 20.500

Brian Lloyd I TUOI AMICI COMMODORE 16 E PLUS 4

Anche i computer hanno un cuore: impara a programmare con i tuoi amici C16 e Plus 4.

cod. 423B Pag. 168
Lire 16.000

Rodney Zaks

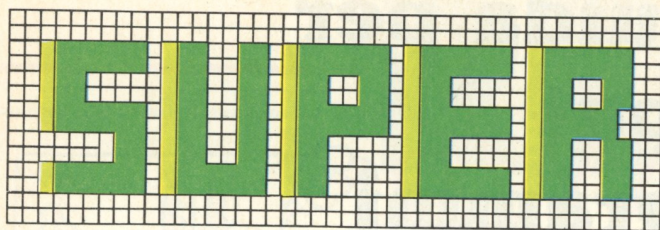
IL TUO PRIMO COMPUTER

Una semplice introduzione al mondo dei personal orientata ad utenti alla loro prima esperienza con il computer.

cod. 351D Pag. 240
Lire 25.000



**GRUPPO
EDITORIALE
JACKSON**



5 EDITORIALE **POINT**

a cura di **Pietro Dell'Orco**

7 **READ & WRITE**

LA POSTA DEI LETTORI

10 **OPEN**

SEMPRE LO STESSO TASTO?
di **Marcello Spero**

14 **REM:HW**

COME TI MIGLIORO LO SPECTRUM
di **Marcello Spero**

17

IL DOMANI DEL SOFTWARE QL

18 LA CULTURA **then** INFORMATICA

AVVENTURA NELL'ADVENTURE
di **Brian Robb**
trad. e adatt.
di **Lucio Bragagnolo**
Seconda parte

22 **RANDOMIZE**

L'ASSEMBLY ASSIEME
di **Marcello Spero**
Quinta parte

30 **LOAD**

30 VIDEOCALC
di **Paolo Bain**

35 R'BERT
di **Roberto Corona**

43 REVERSI
di **Giorgio Simonetti**

48 **QUARK**
di **Luca Ciarlatani**
e **Stefano Di Paoloo**

53 **INTEGRAZIONE NUMERICA**
di **Giuseppe Voglino**

59 **LATINO**
di **Ferruccio Mandorli**

66 **CLASSIFICHE DEI CAMPIONATI**
di **Gianni Mariani**

71 **NURSEY**
di **Luciano Prelli**

75 **PRINT**

RECENSIONI LIBRI
di **Lucio Bragagnolo**

78 **ZX CLUB**

I CLUB SINCLAIR

J. soft s.r.l.
DIREZIONE, REDAZIONE, AMMINISTRAZIONE
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228-68.37.97
68.80.841/2/3

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE:
Lucio Bragagnolo
Marcello Spero

HANNO COLLABORATO A QUESTO NUMERO
Paolo Bain, Roberto Corona
Stefano Di Paolo, Luca Ciarlatani
Ferruccio Mandorli, Gianni Mariani
Luciano Prelli, Giorgio Simonetti
Giuseppe Voglino

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Toffolatti

FOTOCOPOSIZIONE:
Graphotek - Via Astesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonaiti

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n° 199
del 14.04.1984

STAMPA:
Grafiche Pirovano
San Giuliano (MI)

PUBBLICITÀ
Concessionaria per l'Italia e l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 Milano
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500
(11 numeri con cassetta) L. 38.500
(11 numeri senza cassetta) per l'estero:
L. 110.000 (11 numeri con cassetta)
L. 50.000 (11 numeri senza cassetta)
I versamenti devono essere indirizzati a:
J. soft s.r.l.
Viale Restelli, 5
20124 MILANO

mediante emissione di assegno bancario, vaglia o utilizzando il c/c postale n. 19445204.
Per i cambi di indirizzo indicare, oltre al nuovo, anche l'indirizzo precedente ed allegare alla comunicazione l'importo di L. 500, anche in francobolli
© TUTTI I DIRITTI DI RIPRODUZIONE O TRADUZIONE DEGLI ARTICOLI PUBBLICATI SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO TRATTI DALLA RIVISTA ZX COMPUTING - ARGUS SPECIALIST PUBLICATION LTD
1 GOLDEN SQUARE LONDON W 1R3AB



Rivista associata all'Unione Stampa Periodica Italiana

Piccola guida per l'input dei programmi

I programmi pubblicati da SUPERSINC sono stati accuratamente provati e verificati: in questo modo speriamo di avere ridotto al minimo la possibilità di errori. Nel caso comunque un programma dovesse presentare malfunzionamenti, pubblicheremo le modifiche da approntare in uno dei numeri immediatamente successivi della rivista.

Per utilizzare i programmi pubblicati è sufficiente digitare i relativi listati sulla tastiera del calcolatore.

Nel caso dello ZX SPECTRUM, abbiamo previsto una semplice decodifica dei tasti da digitare nel modo grafico, per semplificare la comprensione dei listati.

Come è noto, lo SPECTRUM è provvisto di 2 serie di tasti grafici: una prima serie di 16 caratteri grafici predefiniti (i tasti numerici da 1 a 8 e gli stessi "shiftati") e una serie di caratteri definibili dall'utente nell'ambito di un programma (le lettere da A a U).

In entrambi i casi per ottenere i caratteri desiderati occorre entrare in modo grafico (cursore contrassegnato dalla lettera G lampeggiante), premendo contemporaneamente i tasti CAPS-SHIFT e 9.

Nei nostri listati i caratteri grafici predefiniti sono indicati da una Q seguita dal numero corrispondente al testo che occorre digitare, il tutto racchiuso tra due parentesi graffe.

Ad esempio {G4} significa che occorre digitare il tasto 4, con il cursore in modo grafico.

Analogamente la codifica SG, seguita da un numero da 1 a 8, significa che occorre digitare il relativo tasto numerico, premendo contemporaneamente il tasto CAPS-SHIFT.

Ad esempio quando, si trova la codifica {G2} occorre premere il tasto 2 contemporaneamente al tasto CAPS-SHIFT, ovviamente con il cursore in modo grafico. In entrambi i casi precedenti, quando un simbolo grafico deve essere digitato più volte, i caratteri G o SG della codifica sono preceduti da un numero che specifica quante volte va premuto il tasto grafico indicato.

Così ad esempio {8G5} significa che il tasto grafico 5 va digitato 8 volte e analogamente {4SG1} indica che il tasto grafico 1 e CAPS-SHIFT devono essere battuti 4 volte.

I caratteri grafici definibili (le lettere da A a U in modo grafico) hanno una codifica semplificata: la lettera corrispondente, sottolineata.

Quando in un listato viene presentata, ad esempio, una A sottolineata occorre entrare in modo grafico (al solito premendo contemporaneamente i tasti CAPS-SHIFT e 9) e quindi digitare semplicemente il tasto che contrassegna la lettera A.

Quando leggete	Premete	Vedrete
{G1}	1	
{G2}	2	
{G3}	3	
{G4}	4	
{G5}	5	
{G6}	6	
{G7}	7	
{G8}	8	
{SG1}	CAPS SHIFT 1	
{SG2}	CAPS SHIFT 2	
{SG3}	CAPS SHIFT 3	
{SG4}	CAPS SHIFT 4	
{SG5}	CAPS SHIFT 5	
{SG6}	CAPS SHIFT 6	
{SG7}	CAPS SHIFT 7	
{SG8}	CAPS SHIFT 8	

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

Quando leggete	Premete	Vedrete
<u>A</u>	A	Simbolo grafico definito nel programma in uso.
<u>B</u>	B	
<u>C</u>	C	
<u>D</u>	D	
<u>E</u>	E	
<u>F</u>	F	
<u>G</u>	G	
<u>H</u>	H	
<u>I</u>	I	
<u>J</u>	J	
<u>K</u>	K	
<u>L</u>	L	
<u>M</u>	M	
<u>N</u>	N	
<u>O</u>	O	
<u>P</u>	P	
<u>Q</u>	Q	
<u>R</u>	R	
<u>S</u>	S	
<u>T</u>	T	
<u>U</u>	U	

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

E D I T O R I A L E

POINT

Novità

Le novità sono nell'aria.

Dal numero di ottobre, infatti, abbiamo pensato di creare, attorno alle nostre riviste, nuovo e speriamo sempre più ampio interesse.

Vediamo come: prima di tutto la rivista avrà 16 pagine in più; pagine che avranno carattere giornalistico e quindi non listati, ma articoli di vario genere, da leggere per cultura o informazione.

Tra le varie iniziative vi è la realizzazione di due nuove riviste mensili: SuperMSX e SuperApple. Tra voi c'è forse chi occhieggia un sistema più evoluto e uno sguardo alle nuove testate J.soft può sempre essere utile. Ad ottobre parte anche la campagna abbonamenti, tradizionale ed importantissimo evento editoriale per tutte le riviste.

Gli abbonati alle nostre pubblicazioni hanno sofferto un po' in passato, causa un certo disservizio generato da tutta una serie di piccoli problemi, che, sommati, hanno provocato i guai che noi per primi abbiamo lamentato.

Le cose da questo punto di vista si stanno normalizzando e siamo in grado di garantire fin d'ora, per l'immediato futuro e per quanto in nostro potere, un eccellente servizio, poiché la periodicità sarà finalmente rispettata e il servizio abbonati diverrà decisamente più efficiente, grazie a nuove procedure più veloci e più "snelle".

Chiediamo per l'ennesima volta scusa agli abbonati, pregandoli di voler considerare chiusa una spiacevole parentesi.

Il nostro impegno si fa e si farà sempre più concreto, confidando nel costante interesse e apprezzamento che finora ci avete dimostrato ... nonostante tutto!

Pietro Dell'Orco

Gli ultimi saranno i primi?

Giudicalo tu.

Anche se, in ordine di tempo, siamo gli ultimi a uscire con libri "dedicati" ai più diffusi home e personal computer, siamo convinti che il lettore attento ed esigente apprezzerà la qualità dei contenuti e la loro presentazione.

Abbiamo selezionato e tradotto i best-seller mondiali che accolgono quanto di più utile è stato scritto per il tuo computer.

Scegli a colpo sicuro!

Puoi ordinare direttamente i titoli presentati compilando ed inviando il coupon pubblicato oppure acquistarli presso i più qualificati computer shop e le migliori librerie.

CEDOLA DI ORDINAZIONE - LIBRI

Compilare e spedire in busta chiusa a:
J. soft - Viale Restelli 5 - 20124 Milano
tel. 02/6880841-6880842-6880843

Ordino i seguenti libri per un importo totale di L.
+ L. 2.000 come contributo fisso per spese di spedizione

Cod. Cod.
Cod. Cod.
Cod. Cod.

- Contanti allegati
- Assegno allegato n°
- Ho spedito l'importo a mezzo vaglia postale
- Ho versato l'importo sul CCP n° 19445204 intestato a J. soft - Milano
- Pagherò in contrassegno al postino al ricevimento dei volumi

Nome

Cognome

Via

CAP Città Prov.

Se richiesta fattura - codice fiscale

Data

J. soft: libri di qualità

Per Commodore 64

Giochi fantastici per il C 64
Cod. ASOC 001 L. 42.000
(inclusa cassetta)

Un libro pensato e realizzato per il divertimento dell'utente del C 64; contiene 19 giochi di vario genere tutti ampiamente commentati. L'attento studio dei programmi potrà essere vantaggioso per l'apprendimento delle tecniche di programmazione.

Il libro del C 64
Volume 1
Cod. ASOC 010 L. 24.000
(inclusa cassetta)
Il libro del C 64
Volume 2
Cod. ASOC 011 L. 24.000
(inclusa cassetta)

I due libri illustrano a fondo le possibilità del Commodore 64. Un compendio di utili consigli, e quant'altro è necessario per conoscere meglio il proprio calcolatore. Esempi pratici completano l'esposizione in modo chiaro ed esauriente.



C 64: suono e grafica - Volume 1
Cod. ASOC 008 L. 24.000
(inclusa cassetta)
C 64: suono e grafica - Volume 2
Cod. ASOC 009 L. 24.000
(inclusa cassetta)

Due volumi che non possono mancare nella biblioteca dell'utente Commodore. Una miniera di idee e suggerimenti per la programmazione del tuo personal computer. Una vasta serie di programmi esemplificativi guidano il lettore al miglior sfruttamento del Commodore 64.

Per ZX Spectrum

15 giochi grafici per ZX Spectrum
Cod. ASOC 002 L. 24.000
(inclusa cassetta)

Un fantastico volume con 15 entusiasmanti giochi grafici per il tuo ZX Spectrum. Tutti i programmi sono "pronti

all'uso" e ampiamente commentati. I programmi sono scritti sfruttando appieno le capacità del calcolatore ed il loro studio può giovare a chiunque intenda affinare le proprie tecniche di programmazione.



Inoltre, i Paper Book: la raccolta dei programmi suddivisi per macchina di Paper Soft, il primo settimanale di software su carta per il tuo computer.

Paper Book - programmi per Apple II -
Cod. ASOC 005 L. 18.000
Paper Book - programmi per VIC 20 -
Cod. ASOC 003 L. 18.000
Paper Book - programmi per Commodore 64 -
Cod. ASOC 004 L. 18.000
Paper Book - programmi per Texas TI 99/4A -
Cod. ASOC 007 L. 18.000
Paper Book - programmi per ZX Spectrum -
Cod. ASOC 006 L. 18.000

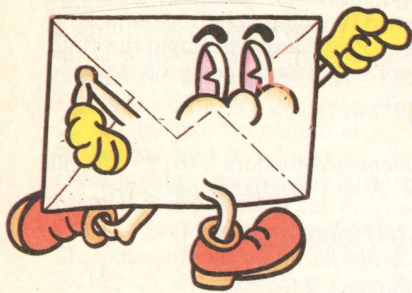


In ogni volume, corredato di cassetta, oltre 40 programmi per il tuo computer: giochi grafici, utility, arcade, adventure una miscela esplosiva di software "pronto all'uso"!

J. soft
S.r.l.

Viale Restelli, 5 - 20124 Milano - tel. 02/6880841-6880842-6880843

READ & WRITE



La posta dei lettori

Sono OUT con e IN...

Carissimo SUPERSINC

ho terminato l'interfaccia, il monitor sonda e l'interfaccia per il joystick descritti da Marcello Spero nelle diverse puntate apparse su SUEPRINC. Sembra che tutto sia OK. Scrivo "sembra" perché in OUT l'insieme funziona a meraviglia, mentre in IN non riesco a farla funzionare con il joystick.

Mi spiego meglio: se inserisco un programma non riesco a utilizzare il joystick con l'interfaccia, mentre con il programma

10 PRINT AT 10,10; IN 31

20 GOTO 10

visualizzano i valori 1-2-4-8-16, rispettivamente con D0-D1-D2-D3-D4.

Vorrei inoltre chiederti se con questa interfaccia posso pilotare la stampante GP 100 A della Seikosha e al DEC printer III della Digital.

Ti ringrazio e aspetto altre novità su SUPERSINC.

G. Lucio Bregoli - Brescia

R. Risponde Marcello Spero:

Carissimo G. Lucio,

ti includo con piacere nel nostro di quanti hanno avuto successo nella realizzazione del nostro progetto di interfaccia.

Se infatti il programmino di verifica ti dà i risultati che dici, si può dire con sicurezza che è a posto anche la sezione di input.

Quanto all'altro problema, direi che non riguarda l'hardware (computer ed interfaccia) ma il software che utilizzi.

Perché un "game" possa rispondere ai comandi di un joystick collegato alla nostra interfaccia, è necessario che disponga dell'opzione "Kempston joystick" (e, ovviamente, che questa sia stata selezionata prima di iniziare ad usare il programma).

Come è stato spiegato nel corso della realizzazione, infatti, la nostra interfaccia è "Kempston compatibile", cioè i suoi segnali sono codificati secondo questo standard.

L'unico standard di interfaccia in grado di funzionare contemporaneamente alla tastiera (ne usa infatti gli stessi indirizzi) è il Protek-Sinclair.

Passando alle possibilità di utilizzo del-

l'interfaccia per pilotare stampanti di tipo parallelo (come sono i due modelli che hai citato), c'è da dire che la cosa è in teoria possibile. Le modifiche da apportare sarebbero comunque non poche.

Il vero problema è comunque la realizzazione del software di interfaccia, che realizzi un adeguato "protocollo" di collegamento secondo lo standard Centronics. L'idea è però interessante, e non è detto che non possa essere sviluppata in una futura puntata di REM.

Wet Jet Silly

Caro SUPERSINC,

sono un ragazzo di 13 anni e ho da alcuni mesi uno Spectrum 48K.

Fra i giochi che ho, quello che mi ha più entusiasmato è stato Jet Set Willy, che è diventato per me una vera e propria fissazione.

Nonostante tutto il mio impegno, infatti, non sono ancora riuscito a trovarne la soluzione.

Ti prego, se possibile, di dare modo al mio stanchissimo Willy di andare finalmente a letto.

Sarebbero molto graditi anche i trucchetti vari per avere qualche omino in più o magari per prendere alcuni oggetti "maledetti".

Un'altra soddisfazione che mi vorrei togliere è di vedere sul video del mio computer una bella spirale. Come si fa?

Tanti complimenti per la bellissima rivista (grazie! N.d.R) e l'altrettanto bella cassetta.

Sperando di leggere su uno dei prossimi numeri questa lettera e, soprattutto, le risposte ai miei quesiti, ti ringrazio e ti saluto di cuore.

**Pietro Maccari
Colle di Val D'Elsa (SI)**

Caro Pietro, scusaci se siamo un po' in ritardo nel risponderti. Per farci almeno in parte perdonare, abbiamo cercato di accontentarti il più possibile. Al capitolo Jet

Set Willy ti forniamo alcune POKE davvero micidiali, da inserire in una riga 5 del programmino caricatore del gioco, comunque subito prima dell'istruzione RANDOMIZE USR 33792, che dà inizio al gioco:

POKE 35899,0 ti dà non solo "qualche", ma addirittura infiniti omini in più a disposizione; anche se l'abbiamo già pubblicata, la ripetiamo, per distratti e ritardatari.

POKE 36477,1 permette a Willy di cadere da qualsiasi altezza senza morire. Non male, eh?

POKE 59900,0 elimina il perverso meccanismo ideato dall'autore del gioco che provoca morti casuali visitando l'attico della villa di Willy.

POKE 35123,0 è veramente favolosa: toglie dal gioco tutti gli oggetti in movimento!

POKE 37874,0 infine, raccoglie automaticamente (!) tutti gli oggetti presenti in ogni stanza visitata da Willy.

Ci pare che sia abbastanza, anche se, ricordalo, Jet Set Willy rimane tutt'altro che facile, anche approfittando di notevoli agevolazioni come quelle riportate.

Venendo alla spirale, non si tratta di una cosa molto difficile: per capirne il funzionamento basterà guardare in azione il programma pubblicato qui sotto.

Ricordati solo che il passo della spirale, richiesto in input, deve essere diverso da 0. Un passo ideale è tra 5 e 10: con 1 l'effetto è più che altro decorativo, mentre 2 è il primo passo utile per riuscire a intuire la spirale.

Il programma non possiede controlli di

```
70 INPUT "passo?", passo: "incremento
  o?", incremento
80 LET base=0
90 LET f=0
100 PLOT 127,84
110 LET base=base+passo
120 DRAW ((NOT f)-f)*base,0,PI
130 LET passo=passo+incremento
140 LET f=NOT f
150 GO TO 110
```

nessun genere, e si bloccherà con un errore non appena la spirale diverrà troppo ampia per essere contenuta nello schermo.

Un televisore da SCARTare?

Sono un vostro nuovo abbonato e come tale approfitto subito per sottoporvi un quesito ed eventualmente leggerne la risposta nella pagina della posta della rivista.

Sono entrato in possesso di un televisore da 10" della Philips, denominato "Manet".

Il suddetto televisore - oltre che ricevere i normali programmi della televisione - può essere utilizzato anche come monitor per home computer.

Ho provato a collegare il mio Spectrum 48K al televisore in modo monitor, non ottenendo però alcun risultato: non sono riuscito a collegarmi.

Prima di effettuare il collegamento ho provveduto a modificare l'uscita TV dello Spectrum, collegando uno spezzone di cavo coassiale a monte del modulatore e la calza a massa, quindi ho collegato la nuova uscita ottenuta ai piedini 20 e 17 della presa SCART del televisore, ma, come ho detto prima, non ho ottenuto nessun risultato. Mi sono rivolto allora al rivenditore per avere spiegazioni, ma, come sempre accade, quest'ultimo non è stato in grado di fornirle; mi rivolgo quindi a SUPERSINC, sperando di ottenere tutte quelle indicazioni e suggerimenti per poter utilizzare il monitor. Allego una fotocopia dello schema elettrico del televisore, per la sola parte dei collegamenti dello zoccolo SCART.

Vincenzo Pilati - Bologna

Purtroppo non siamo in grado di darle un consiglio esaustivo: la procedura da lei seguita è corretta, né l'esame dello schema del televisore ha rivelato possibili malfunzionamenti o difetti di costruzione. L'unico punto non chiaro della lettera è quello in cui lei dice di avere collegato "la calza a massa": il cavo deve essere collegato a uno dei terminali di ingresso del modulatore e la calza all'altro. Se lei ha effettivamente operato in questo modo, l'unico consiglio che possiamo darle è quello di provare a invertire i collegamenti di cavo e calza.

TurboLOAD

Spettabile redazione di SUPERSINC, sono un ragazzo di 14 anni e sono un orgoglioso possessore di uno Spectrum 48K.

Ho un piccolo problema: sono un tipo impaziente e perdo la voglia di caricare i programmi quando questi sono troppo lunghi. Vorrei chiedervi se si potrebbe escogitare un sistema per rendere i listati più corti, magari aumentando la velocità di registrazione o altro...non so proprio che fare. Vi prego, rispondetemi!

Fabio Ramassotto - Roma

Caro Fabio, capiamo la tua impazienza, ma la troviamo un pochino esagerata. Pensa agli

utenti Commodore: senza routine come il TurboTape i loro tempi di caricamento possono anche triplicarsi rispetto a quelli Spectrum! Questi ultimi sono già tra i migliori (parlando di cassette) e, oltretutto, gli ultimi programmi usciti delle migliori software house inglesi prevedono tutti qualche sistema di compattezza dei dati che consente di ridurre la durata del caricamento. Per parte nostra la tua è un'idea che metteremmo volentieri in pratica, solo che non possiamo farlo, pena una diminuzione di affidabilità delle cassette allegate alla rivista. La soluzione più comoda è quella di affidarsi, dove sia possibile, ai microdrive o a un disk drive (un "Jet Set Willy" su cartuccia parte in una quindicina di secondi): altrimenti non ti rimane che... diventare più paziente.

Jet Set Map

Cara redazione di SUPERSINC, allego a questa lettera una preziosissima mappa di "Jet Set Willy"...

**Gabriele Raccio
Pietramelara (CE)**

Caro Gabriele, la tua mappa è ottima: il suo unico difetto è di risultare piuttosto confusa, al punto che spiegarla sulla rivista sarebbe davvero cosa ardua. Perché non provi a ridisegnarla, detagliandola meglio e rendendola più chiara? Noi intanto aspettiamo con fiducia...

Facciamo cifra tonda

Sono un vostro fedele lettore da tempo ed approfitto dell'occasione per farvi i miei complimenti per la rivista; in più vorrei che mi aiutaste a risolvere un mio quesito.

Nella stesura dei programmi mi trovo a volte a dover eseguire operazioni con numeri decimali del tipo $25457.32^*7 = 178201.24$. Come posso fare per ottenere col mio computer - uno Spectrum 48K - un risultato senza decimali ed arrotondato automaticamente per eccesso o per difetto?

Oppure, nell'eseguire un'operazione come $25457^*7=178199$ come posso fare per ottenere automaticamente un risultato arrotondato alle mille lire, sempre per eccesso o per difetto (cioè per difetto se le ultime cifre vanno da 1 a 500 e per eccesso se l'intervallo va da 501 a 999)?

Massimo D'Ascenzo - Roma

Nel primo caso occorre lavorare sulla variabile che memorizza il risultato dell'operazione nel modo seguente:

- 1) aggiungere 0.5;
- 2) eseguire un'operazione del tipo LET nostra__variabile = INT nostra__variabile.

Come si può facilmente intuire, questa procedura fa sì che tutti i valori decimali superiori a 0.49999999.... "slittino" a

un valore di unità superiore (per esempio $14.7+0.5=15.2$: INT 15.2 è uguale a 15), consentendo in questo modo anche l'arrotondamento per eccesso.

Anche il secondo caso si risolve in modo analogo: basta far diventare decimale la parte del risultato che vogliamo arrotondare. Prendiamo l'esempio riportato più sopra e vediamo come operare:

$$25457^*7=178199$$

Dobbiamo arrotondare 178199 alle mille lire. Allora dividiamo per mille:

$$178199/1000=178.199$$

aggiungiamo 0.5:

$$178.199+-0.5=178.699$$

prendiamo la parte intera del risultato:

$$\text{INT } 178.699=178$$

moltiplichiamo per mille:

$$178^*1000=178000$$

l'arrotondamento è stato eseguito.

Se avessimo dovuto arrotondare alle 100 lire avremmo, nel primo e nell'ultimo passo della procedura, diviso e moltiplicato per 100; per le 10 lire avremmo diviso e moltiplicato per 10 e così via.

Quien sabe?

Spettabile redazione, sono un operatore agricolo della bassa modenese. Possiedo uno Spectrum 48K e vi scrivo per avere notizie relative ad applicazioni dello Spectrum in agricoltura.

Qualora vi fossero delle aziende agricole che ne fanno uso vorrei richiederne, se possibile, l'indirizzo.

Se possibile preferirei una risposta privata, anche perchè l'ambiente è piuttosto restio ad innovazioni di questo tipo. Parlandone in giro, la prospettiva di un impiego del computer in agricoltura suscita diffidenza...

**Davide Cavallari
Via Trento e Trieste 21
41034 FINALE EMILIA MO**

Purtroppo, non sappiamo dirle se effettivamente esistano altri agricoltori interessati all'applicazione dell'informatica in campo agricolo. Pubblichiamo tuttavia il suo indirizzo, così che chi volesse contattarla in merito possa farlo senza problemi. Per parte nostra ci auguriamo che vengano presto superate le diffidenze di cui sopra, tutto sommato espressione di una mentalità per certi aspetti eccessivamente conservatrice, e che comunque appare destinata, nel lungo periodo, a doversi modificare progressivamente, per amore o per necessità.

Ci tenga al corrente delle iniziative di sua conoscenza.

Per collaborare a SUPERSINC

La maggior parte dei seguenti suggerimenti ha lo scopo di migliorare l'accuratezza e la velocità di pubblicazione di un articolo; rispettando questi consigli si accresceranno le probabilità che un vostro lavoro venga pubblicato. La rivista è interessata ad articoli e programmi riguardanti la linea Sinclair. Siamo più interessati al contenuto di un articolo piuttosto che al suo stile, e soprattutto gli articoli devono essere chiari ed esaurienti. La seguente guida permetterà che le vostre buone idee e i vostri programmi vengano più facilmente accettati per la pubblicazione:

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente alla configurazione richiesta (memoria occorrente, eventuali periferiche e così via).

3 il titolo sottolineato dell'articolo dovrà iniziare a due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina progressivo. Per esempio, Horace Goes... /Brambilla/2.

5 tutte le linee del testo dovranno essere battute con spazio 2 o spazio 3, e un margine di circa un centimetro dovrà trovarsi ad entrambi i lati dello scritto.

6 dovrà essere usata una carta formato A4 e lo scritto dovrà occupare un solo lato del foglio (caratteri maiuscoli e minuscoli).

7 i fogli dovranno essere uniti con una clip.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il la cartuccia per Microdrive dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il titolo del-

l'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come suggerimenti di programmazione si consiglia di usare per esigenze di stampa listati, le istruzioni INK, PAPER, INVERSE piuttosto che scrivere direttamente in INVERSE VIDEO. Un rapido controllo dei programmi per operare queste sostituzioni sarà da noi estremamente apprezzato.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT etc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza - da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere fotografie, questi dovranno avere formato 24x36, o 6x6, in bianco e nero o diapositive.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del programma (da un minimo di L. 50.000 a un massimo di L. 300.000). Il pagamento è effettuato in caso di pubblicazione del lavoro.

15 il materiale ricevuto e non pubblicato non verrà restituito.

Spedite i vostri lavori a:

SUPERSINC
Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione



Sempre lo stesso tasto?

di **Marcello Spero**

Esaminiamo insieme la "Saga 1 Emperor", una delle più famose tastiere alternative disponibili per lo Spectrum "non-Plus"...e anche per il Plus

Sono passati ormai più di due anni dalla comparsa dello Spectrum sul mercato italiano, e il successo di questa stupefacente creazione della fertile mente di Sir Clive non accenna a diminuire.

I motivi? Provate a scrivere un elenco di suoi difetti...siete in difficoltà, vero? Con molta probabilità vi sarete bloccati dopo il primo della lista; lo Spectrum ha infatti un unico, grande (imperdonabile?) difetto: la tastiera.

Questo particolare tanto trascurato dai progettisti della Sinclair nel lontano 1982, e ripreso tardivamente in considerazione con l'avvento dello Spectrum Plus (ma purtroppo non ci siamo ancora del tutto...), è stato ed è il grande handicap di questo computer.

Perché cambiare tastiera?

Vediamo brevemente quali dovrebbero essere le caratteristiche di una buona tastiera, e cosa ci offre il mercato.

Diciamo innanzitutto che se i "gommini" dello Spectrum possono aver dato noia a più d'uno nella stesura dei programmi, ci sono altri settori in cui questa carenza è stata determinante per la mancata diffusione di questo computer, per lo meno fino a poco tempo fa.

Un campo di attività da cui lo Spectrum sembrava inesorabilmente escluso è quello dell'elaborazione di testi. La tastiera totalmente inadatta e la poco pratica memoria di massa a cassette audio avevano dissuasato anche i più temerari. Con l'avvento dei microdrive si è resa disponibile una memoria di massa veloce e adatta a questo genere di applicazioni. Per quanto riguarda la tastiera da parte Sinclair non ci è pervenuto alcun



La tastiera appena tolta dalla confezione, completa di accessori e ricambi.

aiuto, almeno fino all'ingresso in commercio del Plus, che d'altra parte non sembra essere adatto ad una battitura molto veloce.

Solo con una tastiera esterna tutta la potenza di programmi come Tasword II in combinazione con Masterfile può essere sfruttata. Per chi non avesse mai sentito parlare di questi due programmi, che vengono offerti in omaggio con l'acquisto dell'interfaccia 1 e microdrive, diremo che si tratta di un word processor e di un data base, che possono funzionare in modo congiunto.

In pratica, è possibile richiamare dal data base (programma di archiviazione) le informazioni che interessano ed inserirle all'interno di documenti o lettere pre-

parati con il word processor. È interessante, in particolare, il cosiddetto "mailmerge". Si tratta, in sostanza, di reperire all'interno dell'archivio tutti i nominativi rispondenti a determinate caratteristiche, ed inserirli in posizione opportuna all'interno di altrettante copie di una lettera standard.

Una lettera di sollecito, ad esempio, sarà inviata a tutti i clienti in ritardo con i pagamenti. Della ricerca dei nominativi opportuni si occuperà il data base, esaminando le informazioni relative ad ognuno, mentre il word processor fornirà una lettera adeguata.

Anologo discorso potrebbe essere fatto per i programmi di calcolo tipo VU-Calc, poco sfruttabili con una tastiera

che rende pazzesca l'introduzione di lunghe serie di dati numerici. Dicevamo delle caratteristiche di una buona tastiera. Quali sono? In effetti non c'è una regola fissa e rigorosa. L'essenziale è che i suoi tasti siano mobili quel tanto che basta ad accompagnare le dita ed a far percepire l'avvenuta pressione. In questo loro movimento dovrebbero essere morbidi, ma opporre una minima resistenza, e non "traballare" assolutamente.

Al di là di queste caratteristiche di base, poco può essere detto in teoria.

Solo la prova pratica può stabilire se una tastiera sia meglio di un'altra.

Certo, un ruolo importante è giocato dall'ergonomia dei tasti, ossia la loro forma e posizione. Una tastiera mal fatta, con i tasti poco accessibili, anche se meccanicamente è perfetta, si farà notare dopo qualche ora di lavoro, quando le mani inizieranno a far male. D'altra parte, una tastiera ben studiata, arcuata, con la barra spaziatrice alla giusta altezza, darà subito una sensazione di grande comodità.

Per lo Spectrum...

Più in particolare, per quanto riguarda lo Spectrum c'è qualche altra considerazione da fare.

La sua tastiera originale è, infatti, un po' troppo "piena". Per contenere il numero dei tasti i progettisti della Sinclair hanno riunito un gran numero di simboli su ciascun tasto.

Di per sé questo non è un gran male; il guaio vero è che per selezionare di volta in volta l'uno o l'altro simbolo occorre premere continuamente uno dei due SHIFT, o tutti e due e poi ancora uno, come tutti ben sappiamo.

Le soluzioni a questa situazione ingarbugliata sono di due tipi.



La tastiera vera e propria. Notare il gran numero di tasti a disposizione e la posizione giocosa dei cursori.

La prima, più sofisticata, prevede l'introduzione di nuovi tasti "autoshiftati". Si tratta di tasti, premendo i quali viene automaticamente premuto anche lo SHIFT appropriato. Ecco allora apparire un tasto per il punto, uno per la virgola, uno per il DELETE, uno per il modo EXTENDED, i tasti cursore.

È intuitivo che la velocità e la semplicità d'uso fanno in questo modo un balzo in avanti.

Il sistema appena visto è utilizzato dallo Spectrum Plus.

Il secondo sistema è più semplice, ma non per questo meno efficace.

Si tratta di duplicare i due SHIFT e collocarli vicino ai tasti che ne fanno un

uso frequente. Un passo ulteriore è quello di replicare alcuni dei tasti più usati e porli nelle vicinanze dello SHIFT adatto.

Ecco così i due SHIFT l'uno accanto all'altro, per ottenere rapidamente il modo EXTENDED, una replica del tasto DELETE (lo 0) posta accanto al CAPS SHIFT, i quattro tasti cursore posti a croce con al centro il loro SHIFT.

Un'ultima considerazione va fatta circa la realizzazione meccanica dei tasti.

Quella "ortodossa" (e più costosa) prevede una coppia di contatti sigillati all'interno di una piccola scatoletta alla base di ciascun tasto. I tasti, ammortizzati a molla, vengono quindi fissati su di un'apposita base.

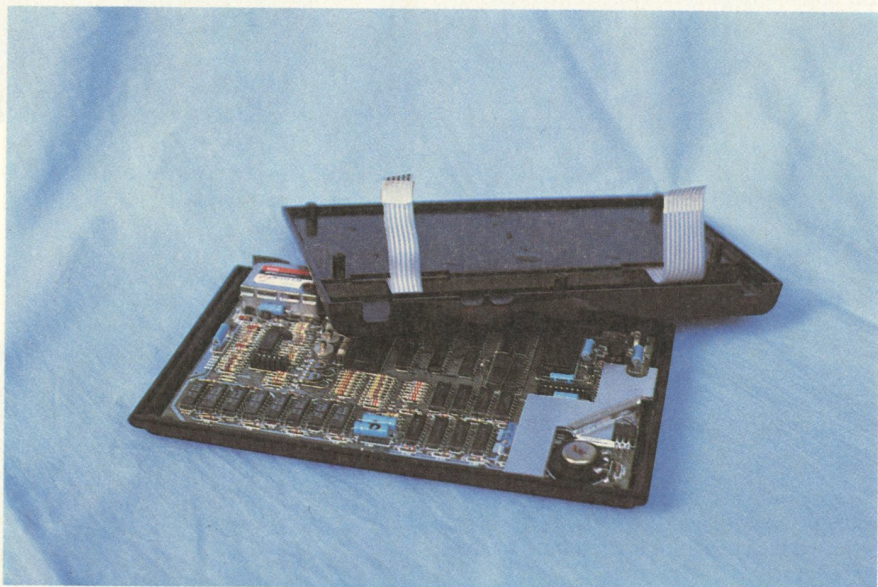
Un sistema più economico è quello che ricalca un po' la tastiera originale Spectrum. I contatti non sono infatti isolati e sigillati, ma disposti su due fogli di plastica flessibile; la pressione dei tasti farà sì che questi due fogli si tocchino, realizzando il contatto elettrico.

Ancora una volta, non è detto che questa soluzione sia più scadente della prima, eccetto che nella durata, decisamente inferiore, dei contatti. Tutto dipende da come è stata realizzata la sospensione dei tasti. È qui, infatti, che "scivola" la tastiera originale.

La Saga 1 Emperor

Dopo aver dato un'occhiata alle caratteristiche generali, eccoci a prendere in esame un prodotto presente sul mercato italiano.

Il suo nome è Saga 1 Emperor, ed è una delle ormai numerose tastiere aggiuntive, da collegare con procedure più o meno complesse allo Spectrum.



Per montare la Saga Emperor bisogna liberarsi della vecchia "carne morta"...

Di cosa si tratta

Prodotta dall'inglese Saga Systems Limited, 2 Eve Road, Woking, Surrey, ed importata da ATW Studio s.n.c. (via dei Pestagalli 7 - 20138 MILANO MI - tel. 02/502204), la SAGA 1 EMPEROR è una tastiera di dimensioni standard, ad elevato numero di tasti (ben 67), dal profilo estremamente basso, secondo i più moderni dettami ergonomici. I suoi tasti hanno una corsa di circa 4 mm, con un lieve scatto che indica l'avvenuto contatto.

La "carrozzeria", di colore bianco avorio, ha uno "styling" davvero elegante.

La sua parte inferiore è forata, per permettere l'inserimento dello Spectrum nel modo che vedremo più avanti.

Una sorpresa è data dai simboli sui tasti, che non sono inseriti ad intarsio, come avviene normalmente nelle tastiere professionali, ma applicati per mezzo di una plastica autoadesiva. Sulla validità di questa scelta, probabilmente dettata dalla necessità di disporre di un prodotto intercambiabile fra Spectrum e ZX81, torneremo alla fine della nostra prova.

Per quanto riguarda i nuovi tasti c'è da dire innanzitutto che sono disposti tutti intorno a quelli "classici", che ripetono esattamente, per posizione e funzione, quelli originali. Non si tratta, per rifarci al discorso appena fatto, di tasti "auto-shiftati", ma di ripetizioni dei tasti normali.

I tasti replicati sono quelli che si suppongono di uso più frequente. Ciascun gruppo è disposto attorno ad un CAPS SHIFT o SYMBOL SHIFT, a seconda dell'esigenza. Troviamo così in alto a sinistra un SYMBOL SHIFT circondato dai tasti +, -, *, =, # e / (quest'ultimo purtroppo riporta la dicitura CLS). Si tratta, come abbiamo detto, dei normali tasti LIST, LOAD, BORDER, 3 e CLS; il loro uso "shiftato" risulta però di gran lunga più agevole, data la presenza centrale dello shift adatto.

Più in basso ecco CAPS LOCK, GRAPHICS, DELETE ed EDIT intorno ad un CAPS SHIFT. Sul lato destro altri due SYMBOL SHIFT: uno nella posizione normale, l'altro circondato da virgola, due punti e punto e virgola. Pure vicini a uno shift si trovano i tasti cursori, due a destra e due a sinistra.

Per coronare questa abbondanza troviamo uno zero supplementare, un punto proprio di fianco al tasto originale corrispondente (la M) e ben due ENTER, uno sopra l'altro.

Una particolare nota di plauso va alla barra spaziatrice, di dimensioni generose, perfettamente allineata con gli altri tasti e dal movimento davvero perfetto.

Osservando la faccia inferiore della tastiera, notiamo che i suoi contatti sono disposti su due fogli di plastica flessibile, come quelli della tastiera originale.

Il sistema di ammortizzamento, piuttosto originale e senz'altro perfettamente

riuscito, si basa su piccoli coni di gomma, con funzione analoga a quella delle molle nelle tastiere "ricche". È proprio la gomma, anzi, a realizzare quel "click" che è forse la caratteristica migliore della SAGA 1 EMPEROR.

Da notare l'incernieramento della barra spaziatrice, realizzato nel modo più sofisticato, con staffa metallica stabilizzatrice.

Visto comunque che si tratta di un prodotto fornito, per ovvie ragioni, in forma di kit, vediamo insieme cosa bisogna fare per passare dalla scatola alla tastiera funzionante, e che difficoltà possono sorgere.

Proviamo...

Apprendo la confezione la prima cosa che balza agli occhi è la tastiera, snella ed elegante. La scatola, però, non contiene solo questo. Guardando con più attenzione scopriamo uno spezzone di cavo piatto provvisto di due connettori alle estremità, un connettore maschio-maschio, una bustina di viti e dadi e, dulcis in fundo, un sacchetto contenente un'elegante "copertina" per proteggere dalla polvere la tastiera quando non viene usata.

Scavando ancora, troviamo un foglio di plastica autoadesiva, sagomata a quadratini e riportante i vari simboli da applicare sui tasti; si tratta evidentemente di un ricambio. Ecco infine il foglio delle istruzioni, che iniziamo subito a leggere.

Con britannica precisione, per prima cosa sono elencati gli strumenti necessari all'installazione della nuova tastiera, che sono semplicemente due cacciaviti, uno piatto e uno a croce.

Seguendo i passi indicati, per prima cosa provvediamo ad aprire il nostro Spectrum, svitandone le viti alla base. Quindi, con estrema delicatezza, sfiliamo dalle loro sedi i due contatti multipli su banda di plastica, che collegano la tastiera originale al resto del sistema.

Passiamo adesso allo smontaggio della nuova tastiera, allo scopo di prepararla all'"innesto". Per questo è necessario prima di tutto rimuovere le due viti che ne bloccano l'involucro, e quindi separare la tastiera vera e propria dalla base in plastica.

La base va ora avvitata saldamente allo Spectrum, per mezzo delle quattro viti fornite. I relativi dadi vanno tenuti in posizione con due dita.

Ora possiamo inserire i contatti della nuova tastiera nei connettori (quelli da cui avevamo precedentemente tolto i contatti della tastiera originale). Segue il bloccaggio della tastiera alla sua base e la chiusura dell'involucro.

L'intero procedimento a noi ha richiesto all'incirca dieci minuti, e non è escluso che si possa eseguire in un tempo anche inferiore.

Da segnalare sono due piccoli inconvenienti, peraltro facilmente eliminabili. Prima di tutto una delle due piste in plastica recanti i contatti da inserire è

risultata troppo larga, ed ha richiesto una "rifilatura" a mezzo forbici.

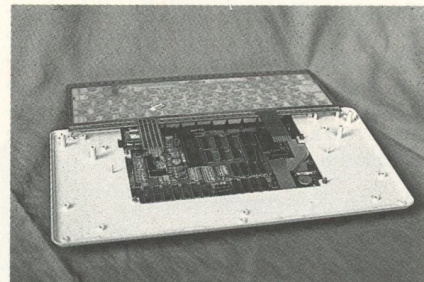
In seguito, avvitando la tastiera alla sua base, due delle viti che fissano lo Spectrum premevano contro la sua parte inferiore, causando falsi contatti.

Anche qui si è resa necessaria una "potatura", questa volta a mezzo pinze.

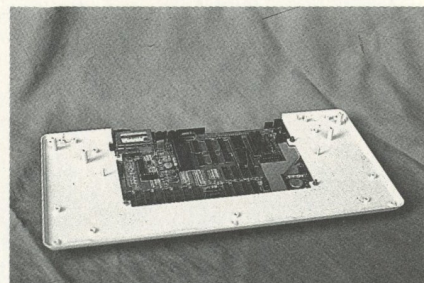
Si tratta, lo ripetiamo, di inconvenienti che nulla tolgono alla validità del prodotto.

Funziona!

Ed eccoci al risultato finale. L'ibrido Saga-Spectrum è stabile e solido, correttamente inclinato e comodo da usare. Confessiamo di essere rimasti un attimo sconcertati, di fronte a tutti questi tasti.



Ecco come la scheda madre dello Spectrum va alloggiata nel suo nuovo involucro.



Collegiamo la nuova tastiera alla scheda madre...



...assicuriamo la tastiera...

Fatta la mano, comunque, la sensazione è quella di avere a che fare con un'altra macchina, dal prezzo ben più elevato. I tasti si fanno trovare da soli, ed il loro "click" sotto le dita dà sicurezza e velocità.

La nostra prova si è svolta su due terreni: la scrittura di un programma e la battitura di un brano di testo, utilizzando il ben noto word processor Tasword II, cui abbiamo accennato all'inizio.

Nel primo caso la velocità è a dir poco raddoppiata, grazie soprattutto alla possibilità di compiere con una sola mano tutte quelle operazioni (EDIT, DELETE, passaggio ai modi GRAPHICS ed EXTEND) che prima ne richiedevano due.

Per quanto riguarda l'uso dello Spectrum con un word processor vale quanto detto nell'introduzione, tenendo presente che la risposta dei tasti è quasi ideale.

Fastidiosa invece, in entrambe le prove, la superficie della plastica autoadesiva che ricopre i tasti. Peccato davvero, visto che il materiale dei tasti offrirebbe invece una sensazione ottima, come dimostra la barra spaziatrice, unico tasto non rivestito.

A proposito di tasti, una segnalazione. L'esemplare della nostra prova presentava uno strano inconveniente: il carattere STOP (A e SYMBOL SHIFT) non veniva assolutamente inviato allo Spectrum, pur essendo perfettamente funzionanti sia la A che lo SHIFT. Pur certi che si tratta di un inconveniente del solo

nostro esemplare, lo segnaliamo perché sia motivo di verifica da parte dell'importatore.

Non è mancata una prova di collegamento di interfaccia 1 e microdrive. Il loro inserimento non ha dato alcun problema, ed anzi la maggiore inclinazione dell'insieme è da considerarsi positiva.

Gli accessori

Se l'interfaccia 1 non dà alcun problema, altrettanto non si può dire per il mare di periferiche in commercio.

Se la vostra interfaccia per joystick non può essere inserita direttamente al connettore dello Spectrum, comunque, non perdetevi d'animo. Proprio per questo sono stati inclusi nella confezione il tratto di cavo piatto con i due connettori e il convertitore maschio-maschio.

Per prima cosa collegate il cavo al connettore di espansione dello Spectrum.

Quindi inserite nell'altro connettore, rimasto libero, il convertitore, che permetterà l'innesto con la periferica. Tutto questo ha il vantaggio di eliminare i rischi di accidentali distacchi delle periferiche, dovuti a un effetto leva con lo Spectrum.

In tema di accessori non si può trascurare la copertina protettiva fornita con la tastiera.

Da notare la sua qualità incredibilmente elevata, vista la funzione relativamente "vile" cui deve assolvere. Per quanto riguarda la sua utilità c'è da dire che la tastiera non richiede una protezione vera e propria contro la polvere, essendo

completamente priva di aperture. Tutto quello che può succedere è che la polvere si infili fra i tasti.

Ben più bisognosa di protezione è invece la parte posteriore dell'insieme, ed in particolare il punto di connessione Spectrum-Saga, dove la polvere potrebbe penetrare all'interno dello Spectrum.

Buono o cattivo?

Per concludere, ci troviamo di fronte a un prodotto con molte caratteristiche positive e qualche difetto.

Fra le prime annoveriamo senz'altro l'estetica, non comune, e il "tocco" quasi perfetto dei tasti, che dovrebbe far meditare i costruttori di prodotti ben più costosi.

Primo fra i difetti è invece il metodo usato per contrassegnare i tasti, che ne rovina l'ottima superficie, oltre ad essere di breve durata. L'inutilità di qualche tasto (lo 0 aggiuntivo, ad esempio...) e le imprecisioni nella realizzazione (le viti troppo lunghe, il contatto da rifilare) possono essere considerati difetti minori, vista la fascia di prezzo in cui questo prodotto si colloca.

Nel complesso un valido sostituto alla vecchia tastiera, ormai giunta all'età della pensione.

Ah, dimenticavamo: il giorno che volete "riesumare" i gommini Sinclair non ci sarebbero problemi: la tastiera originale è perfettamente ripristinabile. Ma c'è davvero qualcuno che lo farebbe?



...e tutto è pronto! Come si vede, microdrive e interfaccia 1 sono perfettamente compatibili.

**IL PREZZO
DELLA
TASTIERA
SAGA EMPEROR
È DI
L. 1.050.000
+ IVA
AI RIVENDITORI
VIENE
PRATICATO
DALLA
ATW STUDIO
LO SCONTO
DEL 20%**

REM:HW

Hardware

Come ti miglioro lo Spectrum

di **Marcello Spero**

Trucchi, consigli e accorgimenti per i più intraprendenti ed...esigenti, partendo da chi non si accontenta della solita tastiera

Iniziamo con questa puntata una breve serie dedicata a quanti, pur entusiasti di questa macchina inesauribile che è lo Spectrum, vorrebbero renderlo un po' più "professionale", ossia più adatto ad essere usato anche come strumento di lavoro.

Cosa manca al piccolo Sinclair per poter essere considerato un utile aiuto nelle attività "serie"?

Tre sono i settori in cui si manifestano delle carenze:

- La tastiera è poco adatta a una veloce introduzione di dati.

- La memoria di massa originale, cioè le cassette, è lenta e scomoda.

- Non è prevista la possibilità di un collegamento con stampanti di elevata qualità, cioè non ZX.

Prima di passare alle "nostre" soluzioni, vediamo, punto per punto, come è possibile risolvere questi problemi con quello che ci offre il mercato.

Per quanto riguarda le tastiere aggiuntive, ne parliamo in questo numero in altra sede, le possibilità sono parecchie, a diversi livelli di prestazioni e di prezzi. Teniamo ancora una volta a far notare come anche lo Spectrum Plus, sebbene possieda una tastiera notevolmente migliore del modello precedente, sia scarsamente adatto a un impiego professionale.

La soluzione più tipica al secondo problema è l'acquisto dei microdrive. Si tratta anche della soluzione più razionale, vista l'alta affidabilità dimostrata da questo sistema di immagazzinamento dei dati, la sua discreta velocità e la

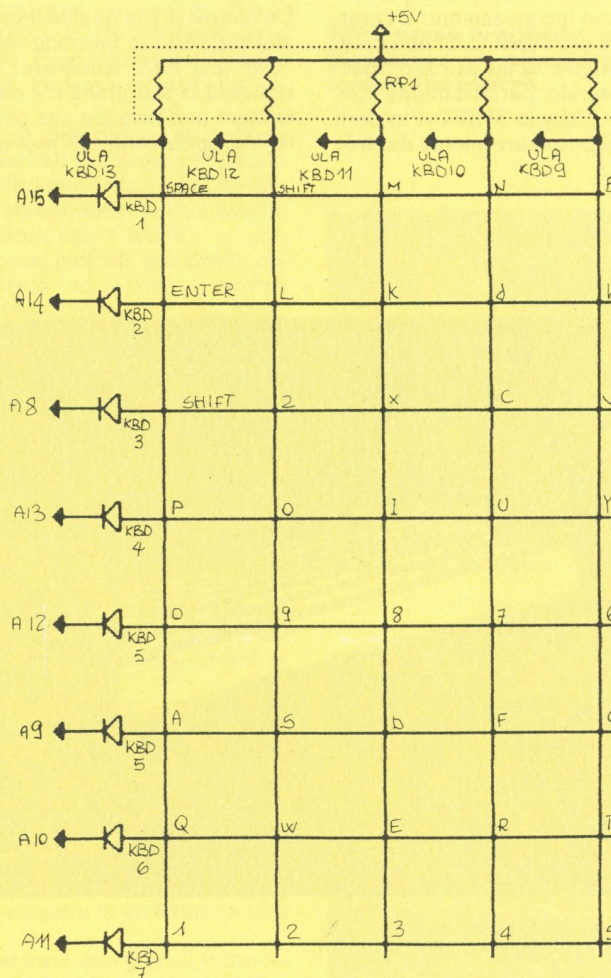


fig. 1: schema della tastiera dello Spectrum

presenza dei relativi comandi già sulla tastiera.

Il prezzo, poi, non è neanche male.

Altre soluzioni possono prevedere una delle ormai numerose interfacce per dischi e un drive di tipo normale, o una soluzione "ibrida" come ad esempio gli ormai famosi (per lo meno in Gran Bretagna) Wafadrive, che ricalcano il funzionamento dei microdrive.

Un altro dei motivi che incoraggiano l'uso, come memoria di massa, dei microdrive, è la possibilità di risolvere contemporaneamente anche il terzo problema, quello della stampante.

L'interfaccia 1, necessaria per il pilotaggio dei microdrive, contiene infatti un'interfaccia RS-232, adatta ad ogni tipo di comunicazione (modem, trasferimento di dati fra una macchina e un'altra di diverso tipo) e anche al collegamento con una stampante, che dovrà essere dotata di interfaccia cosiddetta "seriale".

In alternativa, per chi non possedesse l'interfaccia 1, esistono apposite interfacce per stampante, di tipo seriale, analoghe cioè a quella dell'interfaccia 1, e parallelo, dette anche di tipo "Centronics".

Veniamo al sodo

Dopo questa veloce panoramica riprendiamo ora ciascuno dei tre punti, per una discussione più approfondita e per proporre le nostre soluzioni.

Partiamo dalla tastiera.

Le caratteristiche di una tastiera "seria" dovrebbero essere:

- tasti di tipo professionale, che diano una chiara sensazione quando sono premuti;

- loro disposizione per quanto possibile standard, sia per posizione che per distanza l'uno dall'altro;

- aggiunta di nuovi tasti, per poter effettuare con un'unica pressione tutte quelle operazioni che sullo Spectrum richiedono più pressioni di tasti diversi.

In sostanza, una tastiera tipo quella del Plus, ma con tasti professionali e disposizione più "spaziosa" sarebbe l'ideale. Guardandoci un po' intorno vediamo che simili prodotti esistono sul mercato, ma si tratta di un'elite piuttosto costosa. Non è possibile tentare l'autocostruzione di una tastiera che riunisca tutte le caratteristiche che consideriamo migliori, a un prezzo accettabile?

Noi pensiamo di sì. Oltretutto, una realizzazione di questo tipo potrà essere modellata secondo le personali esigenze di ciascuno, modificando ad esempio la disposizione dei tasti o le loro funzioni.

Noi ci occuperemo prima di tutto dei metodi di interfacciamento di una tastiera esterna con lo Spectrum. Quindi passeremo a vedere come "mettere insieme" una buona tastiera, a partire da quello che si può trovare in qualsiasi negozio di componenti elettronici.

Nostro obiettivo ultimo è mettervi in condizione di realizzare una tastiera

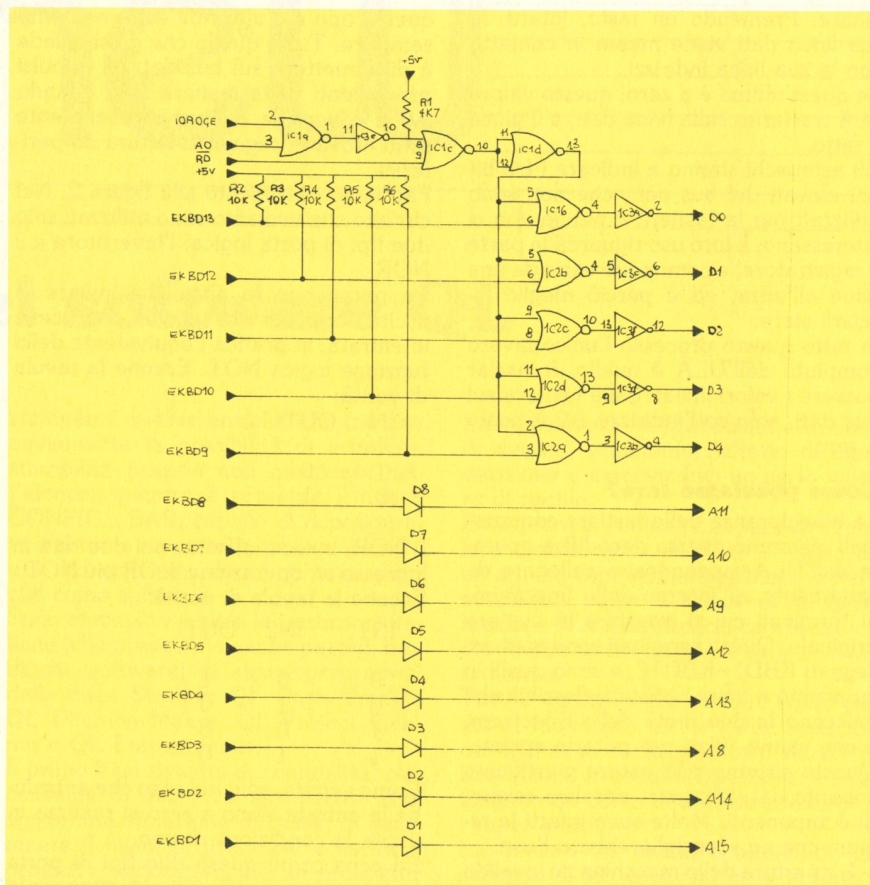


fig. 2: schema del decodificatore per il collegamento di una tastiera esterna

professionale ad elevato numero di tasti, con o senza tastierino numerico, a seconda delle vostre esigenze, e soprattutto collegabile al connettore di espansione dello Spectrum, senza dover smontare la macchina.

Veniamo adesso alla prima parte del problema, e cioè il collegamento di una tastiera esterna con lo Spectrum.

Può sembrare strano trattare per primo questo argomento, ma vi renderete conto in seguito di come l'interfacciamento condizioni di fatto le modalità di realizzazione della tastiera vera e propria.

In principio era la gomma...

Come funziona la tastiera originale dello Spectrum?

I tasti sono organizzati in otto semirighe di cinque, disposte a matrice.

Ciascun tasto, quando è premuto, mette in contatto una determinata linea indirizzi con una linea dati.

L'organizzazione è a matrice. Ogni semiriga ha in comune la stessa linea indirizzi e i tasti, nella stessa posizione in tutte le semirighe, hanno in comune la stessa linea dati.

Le semirighe sono:

1-5	6-0
Q-T	Y-P
A-G	H-ENTER
CAPS SHIFT-V	B-SPACE

In ciascuna semiriga viene considerato

primo elemento il più esterno, ultimo il più interno.

Facciamo qualche esempio.

La "Q" e la "P", entrambi primi elementi di due semirighe, avranno in comune la linea dati D0, ma saranno collegati a differenti linee indirizzi: la "Q" ad A10, la "P" ad A13.

Le linee indirizzi collegate sono quelle che vanno da A8 ad A15, utilizzate in un ordine piuttosto strano, a partire dalla semiriga in basso a destra per finire con quella in basso a sinistra, come vedete in figura.

Normalmente è l'ULA a gestire la lettura della tastiera.

Quando questo circuito riceve il segnale di lettura, costituito da uno zero sulla linea indirizzi A0, esso invierà alla CPU sul bus dati un byte corrispondente allo stato dei tasti di una semiriga determinata dallo Z80, inviando uno 0 sulla linea indirizzi corrispondente. In questo byte saranno a zero le linee corrispondenti ai tasti premuti, e a 1 le altre.

Così, ad esempio, se premiamo il tasto "A", la CPU riceverà sul bus dati il valore ***11110, in risposta a uno zero inviato sulla linea A9, corrispondente alla semiriga A-G.

Per tutte le altre semirighe il risultato sarà ***11111.

Il sistema adottato per portare a zero le linee dati corrispondenti ai tasti premuti è il più semplice che si possa imma-

ginare. Premendo un tasto, infatti, la sua linea dati viene messa in contatto con la sua linea indirizzi.

Se quest'ultima è a zero, questo valore sarà trasferito sulla linea dati, e il gioco è fatto.

Gli asterischi stanno a indicare i tre bit più elevati del bus dati, che non sono utilizzati per la tastiera e perciò non ci interessano. Il loro uso riguarda in parte il registratore, e comunque varia da una issue all'altra, ed è perciò meglio lasciarli stare.

In tutto questo processo l'unico lavoro compiuto dall'ULA è quello di lasciar passare i valori inviati dalla tastiera sul bus dati, solo se l'indirizzo A0 è posto a zero.

Come possiamo fare?

La maggioranza delle tastiere commerciali utilizzano questa decodifica operata dall'ULA, andandosi a collegare direttamente, all'interno dello Spectrum, ai terminali cui si inserisce la tastiera originale. Questi terminali sono contrassegnati KBD1-KBD12, e sono quelli in cui vanno a finire i contatti flessibili che uniscono le due metà dello Spectrum, e che vanno sfilati per poterlo aprire. Questo sistema può essere giustificato soltanto dal risparmio, peraltro esiguo, di componenti. Molte sono infatti le ragioni che ne scongiurerebbero l'uso.

- L'apertura della macchina ne invalida la garanzia, qualora questa sia ancora in atto.

- Per poter stabilire il contatto con i terminali KBD, lo Spectrum deve restare senza la sua parte superiore, perdendo la sua estetica originale e rischiando di "ingoiare" molta polvere.

- Non è possibile l'uso alternato delle due tastiere, ed è sconsigliabile il loro frequente scambio, pena il rapido logorio dei terminali.

- Non essendo facile né rapida la sconnessione della tastiera aggiuntiva, si viene a perdere la portatilità dello Spectrum. Una tastiera professionale è infatti, per sua stessa natura, piuttosto ingombrante.

Per tutti questi motivi, più altri che emergeranno in seguito, per la nostra realizzazione ci orienteremo verso una tastiera con decodifica, da poter collegare al connettore di espansione.

La costruzione di un decodificatore di

questo tipo è d'altronde estremamente semplice. Tutto quello che gli si chiede è di immettere sul bus dati gli impulsi provenienti dalla tastiera solo quando A0 è a zero, e contemporaneamente sono attivati i segnali di lettura da periferica.

Facciamo riferimento alla figura 2. Nel circuito che vediamo sono utilizzati solo due tipi di porta logica: l'invertitore e il NOR.

La prima non fa altro che inviare in uscita l'opposto del segnale che riceve in entrata, in pratica l'equivalente della funzione logica NOT. Eccone la tavola di verità:

IN	OUT
1	0
0	1

Il NOR, invece, effettua sui due dati in ingresso un'operazione di OR più NOT. Eccone la tavola di verità:

IN1	IN2	OUT
0	0	1
0	1	0
1	0	0
1	1	0

Come vedete, solo nel caso che entrambe le entrate siano a zero si realizza in uscita la condizione di uno.

Ed ecco come questi due tipi di porta logica vengono utilizzati per ottenere la decodifica di cui abbiamo bisogno.

Seguiamo la figura.

Per prima cosa vengono letti i segnali IORQGE.CAN# (indica la richiesta di periferica) e A0. Se entrambi sono attivi, cioè portano un segnale zero, l'uscita della prima porta, un NOR, passerà a uno. L'invertitore che segue trasformerà questo zero in uno zero.

Questo zero viene ora unito al segnale RD.can# (richiesta di lettura), e la NOR che compie quest'operazione darà in uscita un livello uno, solo se entrambe le entrate sono a zero. Segue un NOR collegato in modo da svolgere la funzione di invertitore, che trasforma l'uno in zero. Questo "adattamento" è necessario per sfruttare tutte le porte presenti all'interno di un circuito integrato, invece di usarne molti solo parzialmente.

Il risultato di questa prima catena sarà quindi zero, solo se A0, IORQGE# e RD.can# sono tutti a zero. Questa configurazione è inviata dalla CPU per dire: "Voglio leggere la tastiera".

L'impulso di zero viene quindi trasferito a una delle entrate di cinque NOR.

Le loro altre cinque entrate ricevono gli impulsi provenienti dalla tastiera.

Vediamo i vari casi:

1-Le entrate di controllo (quelle cioè che ricevono il segnale in uscita dalla catena di decodifica) sono a uno.

Risultato: al variare degli impulsi provenienti dalla tastiera le uscite resteranno sempre a zero.

2-Le entrate di controllo sono a zero. Risultato: al variare degli impulsi provenienti dalla tastiera le uscite presentano l'opposto del loro valore.

Gli invertitori posti dopo le cinque uscite completano l'opera, trasformando rispettivamente il caso 1 in un livello uno stabile e il caso 2 nello stesso livello letto dalla tastiera.

Abbiamo così ottenuto un dispositivo che trasferisce gli impulsi che riceve solo in presenza di una particolare combinazione di segnali, altrimenti tenendo le sue uscite a livello logico uno. Questo è esattamente ciò che fa l'ULA.

Per le linee indirizzi non ci sono invece problemi. Basterà interporre fra loro e la tastiera dei diodi, che vedete nella figura, per evitare che il segnale possa, in determinate condizioni, percorrere in senso inverso, creando confusione nella lettura.

Sul lato destro della figura vedete i contatti del bus dati e del bus indirizzi, che andranno collegati in modo appropriato al connettore di espansione, come vedremo. Il lato sinistro porta invece i contatti da collegare alla nostra tastiera, marcati EKBD, a significare "external Keyboard", tastiera esterna.

Le resistenze che vedete collegate, in vari punti, alle linee logiche, sono dette resistenze di "pull-up". Il loro compito è quello di mantenere a livello logico uno le linee in assenza di segnale.

Un discorso di corrente

Un capitolo a parte merita l'alimentazione.

Praticamente tutti i produttori di interfacce sfruttano l'alimentazione interna dello Spectrum a 5 volt, per risparmiare componenti.

È noto, d'altronde, che nella versione 48K, e quindi nel Plus, questa linea di alimentazione è già sfruttata più che a fondo. Il collegarsi ad essa con altri circuiti avidi di energia è a dir poco criminale, e quantomeno porta a un riscaldamento oltre ogni limite dell'alimentatore, unito al rischio di irregolarità nella tensione, che possono mandare in crash la macchina.

D'altra parte, per un'applicazione modesta come quella che stiamo esaminando non è il caso di acquistare un trasformatore e collegarsi in modo indipendente alla rete. Basterà invece sfruttare la linea, molto meno sovraccarica, che porta i 9 volt non stabilizzati, provenienti direttamente dall'alimentatore.

Certo, in questo modo dovremo stabilizzare noi la tensione, ma per questo basta un unico circuito integrato e due condensatori, che potranno essere agevolmente collocati sulla stessa basetta del decodificatore. L'alimentazione avverrà perciò attraverso il connettore di espansione.

Tra il dire e il fare...

La prossima volta vedremo come realizzare in pratica tutto quanto abbiamo fin qui descritto, nonché come mettere insieme la nostra tastiera professionale.

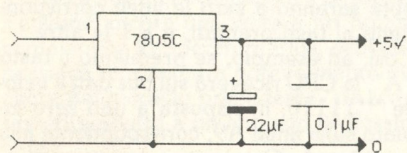


fig.3: schema dell'alimentazione

Il domani del software QL

Conferme e novità nel parco programmi disponibili in Italia per il Quantum Leap

Il QL si colloca per sua natura in una fascia di mercato relativamente atipica: più che sofisticato per essere un home computer, economicissimo rispetto ai personal e business computer, ha provocato qualche incertezza anche all'interno delle software house, prima indecise sul da farsi, vista la scarsa diffusione iniziale delle versioni provvisorie del QL (il cui mercato invece ora in Inghilterra pare "tiri" discretamente) e poi divise tra lo sviluppo di software giocoso o professionale per una macchina in grado di svolgere entrambe le funzioni, ma dalla fascia di utenza non ben definita.

Ora finalmente il mercato software comincia a vivacizzarsi e, sebbene con qualche ritardo, se ne vedono i frutti anche in Italia. Cominciamo dalle versioni 2.0 dei quattro programmi Psion inclusi nel prezzo del QL, ora compattati, più veloci, completamente residenti in RAM all'atto del caricamento e reallizzati interamente in linguaggio macchina. Ecco una tabella di tempi di caricamento e memoria disponibile secondo quanto dichiarato dalla Sinclair:

Tempo di caricamento

PROGRAMMA	VERSIONE 1.0 E SUCCESSIVE	VERSIONE 2.0
Abacus	30 secondi	16 secondi
Archive	37 secondi	17 secondi
Easel	41 secondi	24 secondi
Quill	37 secondi	17 secondi

Memoria disponibile per i dati dell'utente dopo il caricamento

PROGRAMMA	VERSIONE 1.0 E SUCCESSIVE	VERSIONE 2.0
Abacus	15K (1110 celle)	23K (1600 celle)
Archive	12K	20K
Easel	8K	12K
Quill	12K	20K

In più il programma di installazione delle stampanti INSTALL_BAS consente di scegliere immediatamente tra otto

stampanti diverse anziché due (rimane ovviamente la possibilità di installare stampanti proprie non comprese nell'elenco), mentre è presente il nuovo CONFIG_BAS, capace di riconfigurare il sistema sia come hardware (per chi usa, ad esempio, floppy o hard disk) che come software.

Sono imminenti inoltre le traduzioni italiane (che prevedono anche parziali modifiche software) di alcuni programmi della linea Sinclair: QL Cash Trader, QL Decision Maker, QL Project Planner e QL Entrepreneur.

Il primo è un sistema di contabilità, che consente di avere in ogni istante una aggiornata situazione di cassa, di mantenere il bilancio di entrate e uscite e produrre periodicamente un prospetto della situazione IVA. Utilizza le caratteristiche di gestione a finestre e multitasking del QL: il manuale comprende anche una sezione introduttiva.

Project Planner è invece l'equivalente QL del MacProject prodotto per il Macintosh Apple: permette di costruire graficamente il diagramma di un progetto secondo la tecnica Pert, individuando immediatamente le attività critiche del progetto stesso, determinandone il costo e il tempo richiesto e riorganizzando in modo logico l'ordine di ese-

zioni coinvolgenti probabilità e somme di denaro, simulando l'effetto di ogni decisione e associandovi un certo valore di rischio.

Entrepreneur, infine, è capace di simulare diciotto mesi di flusso di cassa di un'attività commerciale all'inizio, determinando il punto di break-even, la situazione IVA e un bilancio complessivo; è possibile inoltre simulare situazioni alternative.

Tutti e quattro i programmi, come già detto, fanno uso esteso delle capacità del QL: sono anche piuttosto simili tra loro quanto a modalità operative, e contengono un corso di autoistruzione per i super inesperti.

La Sinclair sta anche coordinando lo sviluppo di software realizzato in Italia; si tratta prevalentemente di applicazioni professionali. Il tutto dovrebbe essere disponibile al più tardi entro fine anno.

Quanto ai programmi già presenti sul mercato italiano, oltre ai favolosi scacchi Psion, figurano l'Assembler, QL Monitor e QL Toolkit. I progetti futuri comprendono una versione del linguaggio C e del Fortran 77, mentre alcune software house statunitensi stanno lavorando a diversi progetti, che vanno dai programmi per disegnare ai giochi di strategia, a programmi di comunicazione QL/IBM e a un word processor estremamente potente e sofisticato, dotato di un suo proprio dizionario.

Per informazioni rivolgersi a Maria Marchese
Columbia Marketing
Largo Toscanini 1
20122 MILANO MI
Telefono 02/706447-792693

cuzione delle attività. Decision Maker aiuta invece l'utente a eseguire scelte ottimali in un insieme di



Avventura nell'avventure

di **Brian Robb**
 Trad. e adatt.
 di **Lucio Bragagnolo**
Seconda parte

Vediamo insieme alcune subroutine utili a tutti gli scrittori - o aspiranti tali - di avventure su Spectrum e ZX81

Nel primo articolo di questa serie, pubblicato su SUPERSINC del giugno scorso, abbiamo cominciato a spiegare in dettaglio la costruzione del "cervello" di un'avventura, illustrando la struttura di alcune subroutine generali, necessarie ai fini del funzionamento del programma. Per completare il "cervello" il materiale esaminato nella scorsa puntata è però largamente insufficiente; mancano molte subroutine adibite a compiti specifici, alcune delle quali saranno esaminate nel corso di questo articolo.

Per cominciare, vediamo di fare una lista degli oggetti compresi nella nostra avventura, così come nella scorsa puntata ci eravamo occupati dei verbi. I listati che ci occorrono sono i numeri 1 e 2, corrispondenti a quelli intitolati "Elenco parole ZX81" ed "Elenco parole Spectrum" della scorsa puntata. Gli oggetti elencati nei listati sono solo esempi: bisognerà sostituirli con ciò che vorremo mettere nella nostra avventura. Il numero che segue ogni oggetto contraddistingue la locazione in cui questo si trova, e sarà soggetto a eventuali cambiamenti in relazione ai "viaggi" compiuti dall'oggetto durante l'avventura (a seconda che venga preso, trasportato, spostato, lasciato, trafugato e via dicendo).

Sullo Spectrum, a differenza dello ZX81, è necessario disporre di un modulo che legga i dati di oggetti e verbi dalle linee DATA presenti nel programma: entrambe le versioni sono visibili nei listati 3a e 3b.

Ora che il computer possiede anche una lista di oggetti occorre inserire nel nostro programma un modulo che aggiunga alla descrizione di ogni locazione visitata un messaggio che testimoni la pre-



senza dell'oggetto. Il listato 4, valido sia per Spectrum che per ZX81, assolve tale funzione, e praticamente completa il nucleo del "cervello". Manca ancora una routine che, diciamo così, amministri l'esecuzione del programma, chiamando ad ogni input del giocatore la subroutine relativa al messaggio inserito: si tratta del listato 5, anch'esso funzionante per Spectrum e ZX81.

E se parlo difficile?

Abbiamo visto nella prima puntata come la stringa x\$ contenga il verbo inserito dal giocatore e v\$ i verbi facenti parte del vocabolario del computer, in un certo senso la sua "cultura genera-

le". Sempre nel listato 5 si può vedere, alla linea 810, il meccanismo di comparazione che permette al calcolatore di controllare se l'input inserito è di sua conoscenza: vengono esaminati i primi quattro caratteri della frase. Se esiste una corrispondenza con uno dei verbi appartenenti al bagaglio culturale del computer, avviene, come già detto, il salto alla subroutine corrispondente; altrimenti viene visualizzato il messaggio "NON CONOSCO LA PAROLA"; x\$, e toccherà al giocatore digitare un altro comando fino a che troverà un verbo di cui il calcolatore conosce il significato. Il numero dei verbi a disposizione del computer dipende dalla nostra fantasia

e conoscenza della lingua, dalle dimensioni globali dell'avventura e dalla memoria a disposizione. Esistono giochi commerciali con vocabolari di centinaia e centinaia di parole! Tornando in argomento, per poter illustrare meglio il funzionamento di tutto il meccanismo introdurremo ora qualche altra indispensabile subroutine.

Prendi il coltello, presto!

Consideriamo uno dei verbi più usati negli adventure game - PRENDI.IL (o PRENDI.LA o...dipende dall'oggetto). Il listato 6 mostra il modulo di analisi del comando, che adesso spiegheremo.

Supponiamo ad esempio che il giocatore della nostra avventura si trovi nella locazione numero 5, nella quale la nostra perversa immaginazione ha deciso che si debba trovare un coltello. Il giocatore digita "PRENDI.IL COLTELLO", frase che viene spezzata in x\$ - contenente "PRENDI.IL" - e y\$ ("COLTELLO"). PRENDI è il verbo numero 2 della lista, e COLTELLO è l'oggetto numero 1, reperibile nella locazione numero 5. Il programma, una volta analizzata la frase per mezzo del modulo "separazione stringhe" illustrato nella puntata precedente (in cui - approfittiamo dell'occasione per un errata corregge - c'è un errore: in mezzo alle virgolette della linea 690 deve esserci uno spazio, come d'altronde viene anche spiegato nell'articolo), va al modulo che compone il listato 5. Seguiamone l'esecuzione linea per linea:

linea 800: n è uguale a 1; x\$(TO 4) è uguale a "PREN", mentre v\$(1, TO 4) è uguale a "AVAN";

linea 810: le due stringhe sono uguali? Evidentemente no! L'istruzione THEN della linea non viene eseguita;

linea 820: n diventa uguale a 2; x\$(TO 4) è ancora uguale a "PREN", esatta-

mente come v\$(2, TO 4);

linea 810: se le due stringhe sono uguali, allora vai alla subroutine (THEN GO SUB...) v(n).

Dato che v(2) è uguale a 2000, l'esecuzione del programma verrà trasferita alla linea 2000, dove per l'appunto si trova la subroutine di gestione del verbo "PRENDI". È da notare a questo punto che per far riconoscere il verbo al computer è sufficiente digitare "PREN", cosa che, pur non disturbando i novizi, accelererà la velocità di gioco dei più smaliziati ed esperti avventurieri (questa situazione è sperimentabile in molte avventure commerciali: chi ne abbia a casa una provi a troncane qualche comando, digitandone solamente le prime quattro/cinque lettere).

La prima linea della subroutine "PRENDI" (riportata nel listato 6) azzerà la variabile fl, utilizzata come un indicatore. Nell'esempio sopra citato la locazione 1 assume il valore 5, n diventa 1, o(n) diviene 5, y\$(TO 4) è uguale a "COLT", esattamente come o\$(1, TO 4): l'indicatore fl viene allora portato a 1, ad indicare il possesso dell'oggetto, mentre il valore di o(n), solitamente corrispondente al valore della locazione in cui si trova l'oggetto, viene portato a -1, per segnalare al computer il possesso dell'oggetto da parte del giocatore. Visto che fl è uguale a 1, appare il messaggio "OK", dopo di che viene riattivato il modulo del listato 5. Quest'ultimo salta alla linea 100 per ristampare la descrizione della locazione (senza citare l'oggetto preso) e dare modo al giocatore di proseguire l'avventura.

La procedura descritta qui sopra vale anche per la routine di abbandono oggetti, che verrà attivata da un comando come LASCIA.IL COLTELLO: il modulo corrispondente viene mostrato nel listato 7. Anche qui si comincia azzerando fl.

Il programma prosegue successivamente, andando alla ricerca di un valore di o(n) uguale a -1, indicante cioè il possesso dell'oggetto da parte del giocatore. Il calcolatore verifica - confrontando i primi quattro caratteri di y\$ e o\$(n) - che l'oggetto sia quello giusto (ricordate che il giocatore potrebbe possederne più di uno): se tutto va bene, fl assume il valore di 1 e o(n) diventa il numero della locazione in cui si trova il giocatore. L'oggetto rimarrà nella locazione in cui è stato abbandonato, almeno fin quando sarà ripreso. Basandosi sul valore di fl (che è 1) il computer termina di eseguire il modulo stampando un altro messaggio di "OK".

Chiuso per inventario

Un'altra routine indispensabile è quella che stampa una lista di tutti gli oggetti in possesso dell'avventuriero. La potete vedere nel listato 8. Il funzionamento ricalca quello delle altre routine esaminate finora: fl viene riportata per l'ennesima volta a zero, e viene stampato il messaggio "STAI PORTANDO:". Attraverso il ciclo che va dalla linea 4020 alla 4050 il computer prende nota degli oggetti il cui valore, contenuto in o(n), corrisponde a -1, anziché a un numero di locazione, per portare fl a 1 e stampare il nome dell'oggetto. Se o(n) rimane uguale a zero fino al termine del ciclo, significa che l'avventuriero non possiede alcun oggetto; in questo caso fl rimane a zero e viene stampato l'apposito messaggio. Infine il calcolatore ritorna nuovamente ad eseguire il modulo principale del "cervello".

Nella prossima puntata prenderemo in esame alcune subroutine specifiche per l'esempio di avventura che stiamo man mano presentando, contenenti alcune tecniche di uso generale, che vi permetteranno di evitare molti degli inconvenienti più comuni in cui può incappare un aspirante scrittore di adventure game. Arrivederci!

listato 1 inizializzazione verbi versione per ZX81

```

1 REM INIZIALIZZAZIONE VERBI
  ZX81
8200 DIM O$(5,10)
8220 DIM O(5)
8229 REM METTE GLI OGGETTI IN O$
8230 LET O$(1)="COLTELLO"
8239 REM ASSEGNA UNA LOCAZIONE
  A OGNI OGGETTO
8240 LET O(1)=5
8250 LET O$(2)="PISTOLA"
8260 LET O(2)=2
8269 REM ...E VIA DICENDO

```

listato 2 inizializzazione verbi versione per ZX Spectrum

```

1 REM inizializzazione verbi
  Spectrum
8200 DATA "Coltello",5,"Pistola",2:
  REM eccetera, eccetera...

```

listato 3a modulo lettura dati versione per ZX81

```

1 REM MODULO LETTURA DATI ZX81
30 GO TO 8000

```

listato 3b
modulo lettura dati
versione per ZX Spectrum

```
1 REM modulo dati Spectrum
30 RESTORE 8200: FOR i=1 TO 8
40 READ o$(i),o(i)
50 NEXT i
60 FOR i=1 TO (numero dei verbi)
70 READ v$(i),v(i)
80 NEXT i
```

listato 4
routine di stampa oggetti

```
1 REM routine di stampa oggetti
270 PRINT
280 PRINT "PUOI VEDERE "
290 LET FL=0
299 REM 8=NUMERO DEGLI OGGETTI
300 FOR N=1 TO 8
310 IF L=O(N) THEN PRINT O$(N)
320 IF L=O(N) THEN LET FL=1
330 NEXT N
340 IF FL=0 THEN PRINT "NIENTE DI U
TILE."
```

listato 5
routine di analisi verbi

```
1 REM routine verbi
800 FOR n=1 TO (numero di verbi)
810 IF x$( TO 4)=v$(n, TO 4) THEN
GO SUB v(n)
820 NEXT n
830 PRINT "NON CONOSCO LA PAROLA ";x
$
840 GO TO 100
```

listato 6
routine "PRENDI"

```
1 REM routine PRENDI
2000 LET fl=0
2005 REM 8=numero di oggetti
2010 FOR n=1 TO 7
2020 IF l=o(n) AND y$( TO 4)=o$(n,
TO 4) THEN LET fl=1
2030 IF l=o(n) AND y$( TO 4)=o$(n,
TO 4) THEN LET o(n)=-1
2040 NEXT n
```

```
2050 IF fl=0 THEN PRINT "NON E' QUI.
"
2060 IF fl=1 THEN PRINT "OK."
2070 PAUSE 500
2080 RETURN
```

listato 7
routine "LASCIA"

```
1 REM routine LASCIA
3000 LET fl=0
3010 FOR n=1 TO 7
3015 REM 8=numero di oggetti
3020 IF o(n)=-1 AND y$( TO 4)=o$(n,
TO 4) THEN LET fl=1
3030 IF o(n)=-1 AND y$( TO 4)=o$(n,
TO 4) THEN LET o(n)=1
3040 NEXT n
3050 IF fl=0 THEN PRINT "NON POSSIED
I QUESTO OGGETTO..."
3060 IF fl=1 THEN PRINT "OK."
3070 PAUSE 500
3080 RETURN
```

listato 8
routine "INVENTARIO"

```
1 REM routine INVENTARIO
4000 LET fl=0
4010 PRINT "STAI PORTANDO:"
4020 FOR n=1 TO 8
4030 IF o(n)=-1 THEN LET fl=1
4040 IF o(n)=-1 THEN PRINT o$(n)
4050 NEXT n
4060 IF fl=0 THEN PRINT "NIENTE."
4070 PAUSE 500
4080 RETURN
```

Data la brevità dei listati (e, dove non diversamente specificato, la loro compatibilità con Spectrum e ZX81), essi non sono stati registrati sulla cassetta allegata a questo numero.

MISSIONE AFRICA

MIT

Spazio gratuito offerto da J.soft

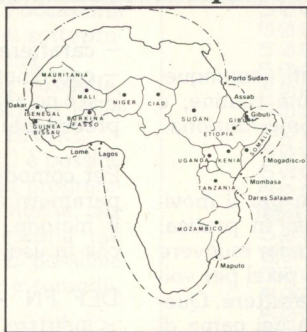
**Una "Nave della Pace" in partenza dall'Italia
porterà in Africa soccorsi immediati e aiuti
per un domani migliore con il vostro contributo.**



La "Nave della Pace": uno strumento nuovo per intervenire immediatamente là dove è necessario un urgente soccorso a popolazioni gravemente colpite da calamità naturali. Non porta solo aiuti immediati ma anche strumenti di lavoro e mezzi ausiliari idonei a realizzare migliori condizioni di vita.

Affinché questo risultato sia raggiunto, gli aiuti siano adeguati ed efficaci e la "Nave della Pace" possa partire a pieno carico occorrono offerte e solidarietà da parte di tutti. Specialisti dei pro-

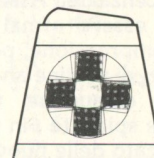
blemi dello sviluppo hanno offerto consigli per la migliore attuazione del progetto. L'iniziativa ha ottenuto l'Alto Patronato del Presidente della Repubblica Italiana, il



Patrocinio dell'ONU, della Lega di Società di Croce Rossa e di Mezza Luna Rossa e della Croce Rossa Italiana e il

contributo del Dipartimento per la Cooperazione allo Sviluppo del Ministero degli Affari Esteri. Il Comitato "Nave della Pace" chiede a quanti comprendono la necessità ed il valore dell'iniziativa, offerte di

mezzi e di beni indispensabili per i soccorsi di prima necessità e per gli aiuti di sviluppo destinati a 15 paesi dell'Africa.



Nave della Pace

COMITATO NAVE DELLA PACE - VIALE MAZZINI 41
00195 ROMA - TEL. 06/317447-386163

Da compilare in stampatello ed inviare in busta chiusa a:
Comitato Nave della Pace
V.le Mazzini 41 - 00195 Roma.

**SI', ANCH'IO VOGLIO AIUTARE
LA NAVE DELLA PACE A PARTIRE
CON LE STIVE PIENE**

Cognome _____

Nome _____

Via _____ N _____

CAP _____ Località _____

Per questo ho deciso di inviare il mio contributo di

- Lit. 10.000 Lit. 50.000
 Lit. 25.000 Lit. 100.000 o più

tramite:

- Assegno non trasferibile intestato:
Comitato Nave della Pace
 C/c postale n. 15285000

- Bonifico bancario a credito
del c/c n. 3100/51 c/o
la Cassa di Risparmio di Roma
sede centrale -
Via del Corso, 320 - 00186 Roma

- Desidero una ricevuta del mio versamento.



RANDOMIZE

L'Assembly assieme

di **Marcello Spero**

Quinta parte

Cominciamo a mettere in pratica quanto imparato finora nel nostro corso di linguaggio macchina

Finite le vacanze? Sia che siate fra i molti che a malincuore devono rispondere di sì, o fra i pochi fortunati che si stanno ancora godendo questo ultimo stralcio di estate, mi auguro che siate pronti ad affrontare una nuova puntata del nostro corso di Assembly. La scorsa puntata (vi ricordate?) aveva segnato l'inizio della fase finale, più evoluta, del nostro viaggio all'interno di questo modo di programmare, a torto ritenuto complicato e pesante. La trattazione classica per argomenti, usata finora, è stata abbandonata, per passare alla presentazione di realizzazioni pratiche.

La nostra conoscenza dell'Assembly, infatti, dovrebbe essere ormai tale da consentirci la comprensione, per lo meno a grandi linee, di routine anche complesse. Queste, comunque, verranno dettagliatamente spiegate, in modo da chiarire il significato delle nuove operazioni che incontreremo. Questo sistema, oltre a farci acquisire una certa familiarità con le tecniche di programmazione usate in Assembly, ha il non trascurabile vantaggio di creare a poco a poco una libreria di routine.

Di volta in volta verrà illustrato l'uso di ciascuna di queste routine. Il loro uso, comunque, non si esaurisce in una loro applicazione separata. Vedremo infatti come unirle in un unico, grandioso videogame, degno degli "arcade" commerciali. Un ulteriore miglioramento a questa nostra creazione verrà apportato durante la seconda fase del nostro corso, che avrà carattere monografico. Ci occuperemo, cioè, di un argomento per volta, che si esaurirà nell'ambito della puntata. Suono, effetti speciali (i cosiddetti stipple - retini -, uso della zona di border) e routine pilotate da interruzioni sono alcuni degli argomenti



che toccheremo.

Prima di correre così avanti, comunque, torniamo alla nostra prima routine, e vediamo com'è fatta e come funziona.

Vi ricordate?

Si trattava di una routine per il movimento ad alta risoluzione. In pratica, con il suo ausilio è possibile far muovere un carattere grafico di un pixel per volta, anziché a scatti di un carattere. Questa tecnica è molto usata nei game di un certo livello, laddove si desidera un movimento fluido e senza sfarfallii.

Della estrema fluidità del movimento prodotto si saranno accorti quanti hanno provato la routine, utilizzando il programma BASIC pubblicato nella scorsa puntata.

Rivediamo comunque, ad uso di quanti avessero perso quella puntata, come si fa ad utilizzare la routine.

I dati di cui questa necessita per un corretto funzionamento sono:

- carattere grafico da visualizzare;
- coordinate di stampa;
- sì o no alla cancellazione della stampa precedente.

Per comodità, il trasferimento di questi parametri è stato realizzato utilizzando il metodo, già ampiamente spiegato, che fa uso di

```
DEF FN <nome> (x,y,z,w) = USR  
<indirizzo>
```

L'indirizzo di inizio, per motivi che vedremo in seguito, deve necessariamente essere 30000. La routine non è infatti rilocabile.

All'atto del richiamo, mediante una

```
<parola chiave> FN <stesso nome>  
(a,b,c,d)
```

i parametri contenuti fra parentesi dovranno essere:

- a: 0 = cancellazione stampa precedente;
1 = non cancellazione;
- b: numero d'ordine del carattere grafico da visualizzare (0 per la "A", 1 per la "B", ecc.);
- c: coordinata x della posizione dell'angolo superiore sinistro del carattere da stampare;
- d: coordinata y dello stesso punto.

Il comando

RANDOMIZE FN q(1,0,0,0)

darà come risultato la visualizzazione del carattere "A" grafico nell'angolo in alto a sinistra dello schermo, mentre

RANDOMIZE FN Q(1,0,248,184);
PAUSE 0

lo riprodurrà in basso a destra. In quest'ultimo caso è necessario, per evitare l'immediata cancellazione del carattere, l'uso di PAUSE. La posizione di stampa utilizzata è infatti nella zona di schermo riservata ai comandi ed ai messaggi, e solo con PAUSE possiamo evitare la comparsa dello

0 OK, 0:1

distruttore.

Abbiamo così chiarito che questa routine permette l'uso di tutto lo schermo, cioè di 256x192 pixel, e che mentre l'asse delle x funziona normalmente, quello delle y risulta capovolto rispetto a quello usato dall'istruzione PLOT.

Un'occhiata d'insieme

Diamo un'occhiata d'insieme al programma; notiamo un blocco principale e un altro denominato "subroutine". Il listato riporta in evidenza questa suddivisione.

Questa osservazione ci dà l'occasione di introdurre l'argomento dei sottoprogrammi in linguaggio macchina.

Anche il linguaggio macchina possiede, infatti, la possibilità di utilizzare delle subroutine, proprio come il BASIC.

Voi tutti sapete che in BASIC è possibile evitare la ripetizione, in un programma, di gruppi di istruzioni identiche, che eseguono elaborazioni necessarie più volte in più punti diversi. È possibile invece, e molto più elegante e comodo, servirsi di una subroutine.

Si tratta di un piccolo (ma non è sempre così) programma, che termina con una istruzione

RETURN

Ogni volta che, nel programma principale, c'è bisogno di eseguire quella particolare procedura, invece di riscriverne le istruzioni che la compongono basterà inserire un'istruzione

GO SUB N

```

***LINEE EQU***
EQU 23728 COOR
EQU 23729 Y_COOR
EQU 23681 CHAR_NR
EQU 23675 UDG
EQU 23563 DEFADD
,
,
,
* * * P R O G R A M M A * * *
Org 50001
,
***ROUTINE PRINCIPALE***
500001 2A 0B 5C      ld hl, (DEFADD)
500004 11 04 00      ld de, 4
500007 19           add hl, de
500008 7E           ld a, (hl)
500009 A7           and a
500010 76           halt
500011 20 07       jr nz, NON
500013 E5           push hl
500014 D5           push de
500015 CD 76 C3    call MAIN
500018 D1           pop de
500019 E1           pop hl
500020 19           add hl, de
500021 19           add hl, de
500022 7E           ld a, (hl)
500023 32 81 5C    ld (CHAR_NR), a
500026 19           add hl, de
500027 19           add hl, de
500028 7E           ld a, (hl)
500029 32 B0 5C    ld (COOR), a
500032 19           add hl, de
500033 19           add hl, de
500034 7E           ld a, (hl)
500035 32 B1 5C    ld (Y_COOR), a
500038 DD E5       push ix
500040 2A 7B 5C    ld hl, (UDG)
500043 EB         ex de, hl
500044 3A 81 5C    ld a, (CHAR_NR)
500047 6F         ld l, a
500048 26 00      ld h, 0
500050 29         add hl, hl
500051 29         add hl, hl
500052 29         add hl, hl
500053 19         add hl, de
500054 E5         push hl
500055 DD E1       pop ix
500057 ED 4B B0 5C ld bc, (COOR)
500061 CD B5 C3   call PIXEL
500064 F5         push af
500065 47         ld b, a
500066 CD CE C3   call PLOT
500069 DD 23     inc ix
500071 ED 4B B0 5C ld bc, (COOR)
500075 48         ld c, b
500076 06 07     ld b, 7
500078 0C         inc c
500079 C5 BC C3     push bc
500080 CD BC C3   call DIS
500083 01         pop bc
500084 0C         inc c
500085 F1         pop af
500086 F5         push af
500087 C5 BC C3   push bc
500088 47         ld b, a
500089 CD CE C3   call PLOT
500092 DD 23     inc ix
500094 C1         pop bc
500095 10 EE       djnz LOOP_M
500097 F1         pop af
500098 DD E1       pop ix
501000 C9         ret

```

dove N è il numero di linea dell'inizio della subroutine.

Le istruzioni contenute in questo "micro programma" verranno eseguite, fino ad incontrare

RETURN

che farà riprendere l'esecuzione dall'istruzione, del programma principale, successiva alla GO SUB da cui la subroutine era stata chiamata.

Che vantaggi ha questo metodo rispetto al salto per mezzo dell'istruzione GO TO? È molto semplice; una subroutine viene di solito utilizzata più volte, nel corso di un programma, e quindi richiamata da posizioni diverse. Come potrebbe fare un'istruzione GO TO a sapere dove tornare?

Una volta sarà

GO TO 20

un'altra

GO TO 100

e così via.

Il meccanismo di richiamo di una subroutine, cioè quello che si serve della struttura

GO SUB

.

.

RETURN

tiene invece nota del numero di linea in cui è avvenuta la chiamata e torna, per mezzo di RETURN, all'istruzione successiva.

Tale numero di linea viene infatti conservato in un'area di memoria chiamata "catasta dei GO SUB". Questo nome deriva dal funzionamento di quest'area, che ricorda da vicino quello di una catasta, o di una pila, di piatti. L'ultimo elemento introdotto sarà infatti il primo a poter essere recuperato. Anche nel caso di chiamate multiple (una subroutine che richiama un'altra subroutine) i ritorni avverranno correttamente nell'ordine inverso alle chiamate.

Tutto quello che abbiamo appena detto per il BASIC vale anche per il linguaggio macchina. Esiste infatti un'apposita operazione di richiamo per subroutine, la cui mnemonica Assembly è

call N

con N indirizzo di inizio, in memoria, della subroutine da richiamare. Questa, a sua volta, sarà conclusa da un'operazione di ritorno:

ret

L'appropriato ritorno è consentito, anche qui, da un sistema di memorizzazione a catasta degli indirizzi di partenza. A proposito, vi siete accorti che questo

```

***SUBROUTINE***
PIXEL
50101 78          ld a,b
50102 CD B1 22    call &22b1
50105 C9          ret

DIB
50106 79          ld a,c
50107 E6 07      and 7
50109 28 02      jr z,OUTCHAR
INC_H
50111 24          inc h
50112 C9          ret
OUTCHAR
50113 7D          ld a,l
50114 C6 20      add a,32
50116 6F          ld l,a
50117 38 F8      jr c,INC_H
50119 7C          ld a,h
50120 D6 07      sub 7
50122 67          ld h,a
50123 C9          ret

PLOT
50124 DD 7E 00    ld a,(ix+00)
50127 0E 00      ld c,0
50129 04          inc b
50130 05          dec b
50131 28 05      jr z,LOAD
LOOP_2
50133 1F          rra
50134 CB 19      rr c
50136 10 FB      djnz LOOP_2
LOAD
50138 AE          xor (hl)
50139 77          ld (hl),a
50140 79          ld a,c
50141 23          inc hl
50142 AE          xor (hl)
50143 77          ld (hl),a
50144 2B          dec hl
50145 C9          ret

```

fig.1: listato completo del programma

ret è lo stesso usato per tornare al BASIC, al termine di una routine in l/m? L'interprete BASIC (che è un programma in linguaggio macchina, posto nella ROM) considera infatti le routine scritte da noi come "subroutine dell'utente" (User SubRoutine, USR), di conseguenza richiamate con lo stesso sistema delle normali subroutine.

Ecco spiegato il perché dei due blocchi; il primo contiene la routine principale, il secondo le varie subroutine, che verranno richiamate al momento opportuno.

Il blocco principale

Iniziamo col prendere in considerazione il blocco principale. Al suo interno possiamo distinguere tre blocchi funzionali:

- lettura primo parametro ed eventuale cancellazione della stampa precedente;
- lettura degli altri parametri;
- esecuzione stampa.

Il funzionamento del primo blocco è molto elementare. Innanzitutto si tratta di portare hl a puntare alla posizione, all'interno della linea DEF FN, in cui è contenuta la prima informazione che ci

serve.

Per questo è necessario partire dall'indirizzo contenuto nella variabile di sistema DEFADD, che indica la posizione del primo parametro, e quindi procedere in avanti di quattro posizioni. Il modo più semplice di fare tutto questo è con un'addizione:

```
ld hl,(DEFADD)
ld de,4
add hl,de
```

Quindi, per influenzare il flag Z con il contenuto dell'indirizzo puntato da hl basterà

```
ld a,(hl)
and a
```

Le operazioni logiche di un operando su se stesso (è il caso di and a) producono sempre un risultato uguale all'operando. L'unico scopo di un'operazione di questo tipo è quello di modificare lo stato dei vari flag, a riflettere il contenuto di a.

A questo punto in caso di

a < > 0

avviene un salto verso l'istruzione etichettata NON, per evitare il blocco relativo alla cancellazione della stampa precedente.

L'operazione

halt

che vedete serve a sincronizzare l'esecuzione del programma con l'elaborazione d'immagine da parte dell'hardware. Per il momento lasciamola stare: ce ne occuperemo approfonditamente più avanti.

Saltando, per il momento, il blocco relativo alla cancellazione, proseguiamo dall'istruzione marcata NON, punto di arrivo del salto.

Eccoci al blocco che recupera gli altri parametri, cioè quelli relativi al carattere e alle coordinate.

La loro sede definitiva sarà in due variabili di sistema normalmente non utilizzate. Il numero d'ordine del carattere andrà all'indirizzo 23681, che abbiamo etichettato, per nostra comodità, CHAR-NR.

Le due coordinate trovano invece posto alle locazioni 23728 e 23729, chiamate da noi rispettivamente COOR ed Y-COOR. Essendo due locazioni adiacenti, i loro contenuti potranno essere trasferiti separatamente in due registri

es.: ld a,(COOR)
ld b, (Y-COOR)

oppure contemporaneamente da una coppia

es.: ld de,(COOR)

indicando semplicemente il primo indirizzo, che per questo non è stato chiamato X-COOR, ma COOR.

Come potete vedere, per passare da un parametro al successivo è necessario aumentare di 8 il valore di hl. Quindi

add hl, de
add hl, de

Dopo aver posto i dati nelle loro aree definitive si entra nel "vivo" della routine.

Le operazioni da compiere sono:

- trovare in che byte è situato, all'interno dell'area video, il pixel individuato dalle coordinate, e a quale bit corrisponde;
- trovare a che byte corrispondono le successive sette linee di pixel che compongono l'intero carattere (il bit sarà sempre lo stesso);
- tranne nel caso in cui il bit in questione sia il numero 7 di un byte, il carattere sarà contenuto in parte nei byte di cui abbiamo trovato gli indirizzi, e in parte nei byte a loro successivi. Per questo occorre preparare, di volta in volta, in due registri i due "pezzi" di ciascuna linea del carattere che andranno nell'uno e nell'altro byte;
- trasferire sullo schermo il frutto di

```
50001 2A 0B 5C ld hl, (DEFADD)
50004 11 04 00 ld de, 4
50007 10 add hl, de
50008 7E ld a, (hl)
50009 A7 and a
50010 76 halt
50011 20 07 jr nz, NON
50013 E5 push hl
50014 D5 push de
50015 CD 76 C3 call MAIN
50018 D1 pop de
50019 E1 pop hl
```

```
NON
50020 10 add hl, de
50021 10 add hl, de
50022 7E ld a, (hl)
50023 32 81 5C ld (CHAR_NR), a
50025 10 add hl, de
50027 10 add hl, de
50028 7E ld a, (hl)
50029 32 B0 5C ld (COOR), a
50032 10 add hl, de
50033 10 add hl, de
50034 7E ld a, (hl)
50035 32 B1 5C ld (Y_COOR), a
```

```
MAIN
50038 DD E5 push ix
50040 2A 7B 5C ld hl, (UDG)
50043 EB ex de, hl
50044 3A 81 5C ld a, (CHAR_NR)
50047 6F ld l, a
50048 06 00 ld h, 0
50050 20 add hl, hl
50051 20 add hl, hl
50052 20 add hl, hl
50053 10 add hl, de
50054 E5 push hl
50055 DD E1 pop ix
50057 ED 4B B0 5C ld bc, (COOR)
50061 CD B5 C3 call PIXEL
50064 F5 push af
50065 47 ld b, a
50066 CD CE C3 call PLOT
50069 DD 23 inc ix
50071 ED 4B B0 5C ld bc, (COOR)
50075 48 ld c, b
50076 06 07 ld b, 7
50078 0C inc c
```

```
LOOP_M
50079 C5 push bc
50080 CD BC C3 call DIS
50083 C1 pop bc
50084 0C inc c
50085 F1 pop af
50086 F5 push af
50087 C5 push bc
50088 47 ld b, a
50089 CD CE C3 call PLOT
50092 DD 23 inc ix
50094 C1 pop bc
50095 10 EE djnz LOOP_M
50097 F1 pop af
50098 DD E1 pop ix
50100 C9 ret
```

fig.2: la routine principale

tutto questo lavoro.

È forse meglio spiegarsi con un esempio.

Supponiamo di voler riprodurre un carattere a partire dalle coordinate x,y. Troviamo che a questa coppia di coordinate corrisponde il bit 4 dell'indirizzo

11. Calcoliamo ora gli indirizzi dei sette byte sottostanti, 12-18, che ospiteranno le righe successive del carattere. Ciò significa che ogni riga del carattere (occupando, come è noto, otto bit) andrà dal bit 4 del byte trovato al bit 5 del byte a lui successivo. L'intera matrice del carattere viene perciò ad essere:

```

PIXEL
50101 78          ld a,b
50102 CD B1 22    call &22b1
50105 44          ld b,h
50106 4D          ld c,l
50107 C9          ret

```

fig.3: la subroutine PIXEL

```

DIS
50108 79          ld a,c
50109 E6 07       and 7
50111 28 02       jr z,OUTCHAR
INC_H
50113 24          inc h
50114 C9          ret
OUTCHAR
50115 7D          ld a,l
50116 C6 20       add a,32
50118 6F          ld l,a
50119 38 F8       jr c,INC_H
50121 7C          ld a,h
50122 D6 07       sub 7
50124 67          ld h,a
50125 C9          ret

```

fig.4: la subroutine DIS

```

PLOT
50126 DD 7E 00    ld a,(ix+00)
50129 0E 00       ld c,0
50131 04          inc b
50132 05          dec b
50133 28 05       jr z,LOAD
LOOP_2
50135 1F          rra
50136 CB 19       rr c
50138 10 FB       djnz LOOP_2
LOAD
50140 AE          xor (hl)
50141 77          ld (hl),a
50142 79          ld a,c
50143 23          inc hl
50144 AE          xor (hl)
50145 77          ld (hl),a
50146 2B          dec hl
50147 C9          ret

```

fig.5: la subroutine PLOT

76543210	76543210		
11	*****	I1+1	***
12	*****	I2+1	***
13	*****	I3+1	***
14	*****	I4+1	***
15	*****	I5+1	***
16	*****	I6+1	***
17	*****	I7+1	***
18	*****	I8+1	***

Vediamo adesso come queste elaborazioni vengono svolte in realtà. Il blocco contrassegnato con MAIN può a sua volta essere suddiviso in tre blocchi funzionali:

- calcolo dell'indirizzo a cui trovare il carattere da stampare;
- trasformazione delle coordinate nel byte e bit corrispondenti, e plottaggio della prima riga del carattere;

- ripetizione per le successive sette righe.

Qualcosa di nuovo

Come prima operazione troviamo subito qualcosa di nuovo:

```
push ix
```

Si tratta di una operazione a noi nuova, push, applicata a un registro, ix, che non abbiamo mai visto.

Iniziamo dall'operazione. Abbiamo parlato poco fa della catasta e di come il sistema se ne serva per conservare gli indirizzi di ritorno delle subroutine.

Questa stessa catasta può servire anche a conservare il contenuto di una coppia. Supponiamo, ad esempio, di avere impegnate tutte le coppie e di doverne liberare una per determinate

operazioni. Come fare a non perdere il suo contenuto?

Potremmo, certo, utilizzare una locazione di memoria come "serbatoio":

```
ld (<indirizzo>), hl
```

Questo comporta però una appropriata scelta delle locazioni, ed è un procedimento poco veloce.

Molto più pratico è

```
push hl
```

che pone in cima alla catasta il contenuto di hl. Per recuperare il valore basterà effettuare l'operazione inversa,

```
pop hl
```

Certo, usando questo sistema occorre tener presente il funzionamento particolare della catasta, che può essere riassunto in:

primo dentro, ultimo fuori

Infatti, nel caso di salvataggi multipli, tipo

```
push bc
push de
push hl
```

al momento del recupero occorrerà procedere in senso inverso:

```
pop hl
pop de
pop bc
```

pena lo scambio dei valori. Come vedremo più avanti, anche lo scambio dei valori può essere una tecnica utile a compiere dei travasi fra coppie.

Torniamo adesso alla nostra routine. Il registro ix, in realtà una coppia, è anche detto, insieme al suo analogo iy, registro indice. Questo perché non può funzionare per le normali operazioni, ma in compenso è in grado di svolgere molto bene la funzione di puntatore.

A noi serve proprio per questo. Vogliamo infatti farlo puntare all'indirizzo che contiene il primo byte del carattere grafico da stampare. Certo, per questo potremmo usare hl, ma preferiamo tenere disponibile quest'ultimo per altre operazioni.

Ma perché salvare il contenuto di ix senza averci ancora messo niente?

Perché al momento di passare all'esecuzione della nostra routine il sistema può avere lasciato in ix informazioni importanti, indispensabili al momento del ritorno dalla routine stessa. Per prudenza, quindi, prima di usarlo ne proteggiamo il contenuto, con l'intenzione di restituirlo al sistema così com'era.

Risolto il problema creato da push ix, passiamo oltre.

Carichiamo, con

```
ld hl,(UDG)
```

in hl il valore di inizio dell'area dei caratteri grafici. Quindi (sorpresa!) trasferiamo il contenuto di hl in de. Avremmo potuto anche caricare direttamente in de il valore, con

ld de,(UDG)

È interessante, invece, conoscere questa operazione, che permette di scambiare rapidamente il contenuto di de e hl. Non esiste un suo analogo per altre combinazioni di coppie. Ora che abbiamo in de l'indirizzo base dell'area UDG, carichiamo in a il numero d'ordine del carattere che ci interessa e procediamo al calcolo del suo indirizzo.

Poiché ciascun carattere occupa otto byte, il secondo sarà in posizione

< base > + 8

il terzo in posizione

< base > + 16

e così via.

In generale, il carattere N sarà in posizione

< base > + 8*N

Occorre quindi moltiplicare per otto il contenuto di a. Ma attenzione! Se il numero d'ordine originale è senz'altro minore di 255 (i caratteri grafici sono infatti molti meno), molto probabilmente non sarà così per N*8. Prima di effettuare la moltiplicazione occorre perciò trasferire N in una coppia. Quindi

ld l,a
ld h,0

E adesso, la moltiplicazione. Il modo più semplice è in questo caso la somma ripetuta:

```
add hl,hl (N*2)
add hl,hl ((N*2)*2)=N*4
add hl,hl (((N*2)*2)*2)=N*8
```

A questo punto basterà sommare al risultato l'indirizzo base

add hl,de

per ottenere il valore desiderato. Ma noi volevamo mettere questo valore in ix. Come fare? Non esiste, infatti, l'operazione

ld ix,hl

Ecco un'applicazione dello scambio di valori a mezzo push-pop:

```
push hl
pop ix
```

e il gioco è fatto.

Passiamo al secondo blocco, quello che calcola l'indirizzo e il bit di inizio della prima riga di carattere e la trasferisce sullo schermo.

Questo blocco si avvale dell'uso di due subroutine, PIXEL e PLOT. Noi, per ora, non ci occuperemo del loro funzionamento, ma continueremo a seguire il programma principale.

Dopo aver caricato in bc le coordinate viene chiamata PIXEL. Al suo ritorno hl contiene l'indirizzo del byte, ed a il numero d'ordine del bit al suo interno, corrispondente alle coordinate date.

Dopo avere provvisoriamente salvato il contenuto di a con

push af

(notate come a sia abbinata al registro dei flag nel salvataggio con push, per formare una coppia) e trasferito in b il contenuto di a (che, pur se già in catasta, resta in a finché ne modifichiamo il contenuto) viene chiamata la subroutine PLOT, che esegue tutto il procedimento di suddivisione della linea di carattere fra due registri e di successivo trasferimento all'area video.

Al suo ritorno ix viene incrementato di 1, in modo che punti al byte successivo del carattere, in bc viene ricaricato il valore delle due coordinate (PLOT modifica il valore di bc), e quindi il contenuto di b (coordinata y) viene passato in c. B è usato come contatore per il ciclo che ripete il procedimento PIXEL-PLOT per sette volte.

La coordinata x, inizialmente in c, va in tal modo persa. Vedremo che il suo valore non ci serve. La subroutine DIS, utilizzata per calcolare gli indirizzi delle linee successive, si accontenta infatti dell'indirizzo della riga precedente, in hl, e della coordinata y, ora in c, che viene incrementata di 1 per indicare la riga successiva.

Come potete vedere, bc è stata salvata prima di chiamare DIS; in questo modo al suo ritorno potremo riaverla integra. Prima di chiamare PLOT abbiamo bisogno di ripristinare il valore di a. Per questo è purtroppo necessario "tirare fuori" anche bc: tanto vale, allora, incrementare c. Dopo il ritorno da PLOT ix viene incrementato per essere pronto alla riga seguente, e quindi bc è recuperato, in modo che b possa funzionare da contatore per il ciclo, che deve ripetersi sette volte. L'operazione

djnz LOOP-M

fa proprio questo.

Al termine del ciclo vengono recuperati i valori di af (inutile, ma bisogna farlo per poter arrivare a ix) ed ix, che, come abbiamo visto, deve avere, al ritorno al BASIC, il suo valore originale.

A proposito di push e pop

A proposito dell'uso della catasta per salvare il contenuto di coppie di registri, è importante ricordare sempre che si tratta della stessa catasta che contiene gli indirizzi di ritorno dalle subroutine. Un routine tipo

```
push bc
push hl
pop hl
ret
```

porterebbe inesorabilmente la macchina al crash. Infatti, al momento dell'esecuzione di ret il valore in cima alla catasta non è quello corretto, ma il contenuto di bc, salvato e non più recuperato. Per cui, servano o non servano, tutti i valori salvati all'interno di una subroutine vanno recuperati prima del ritorno dalla subroutine stessa.

Ciò non vuol dire, comunque, che non sia possibile chiamare una subroutine,

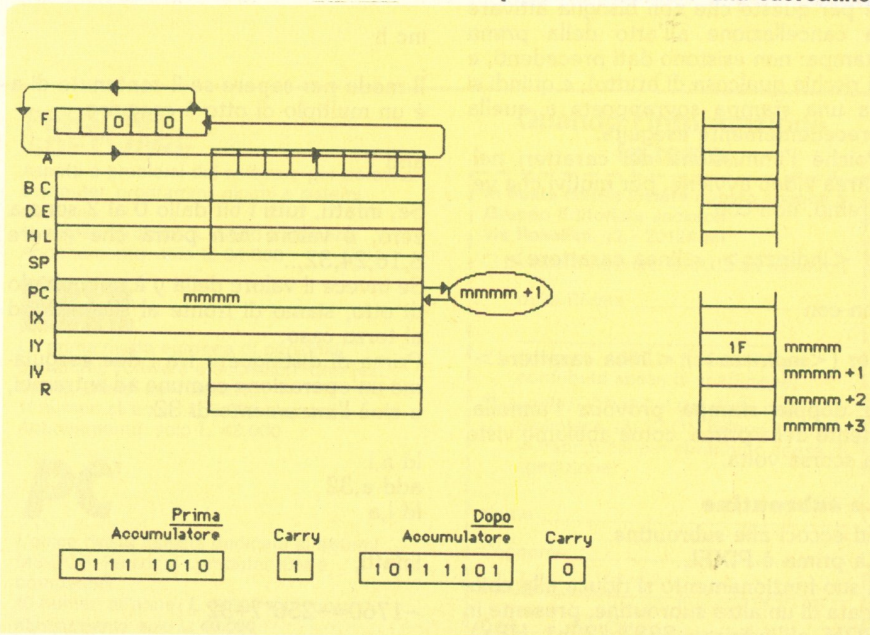


fig.6: schema dell'istruzione di scorrimento RRA

se in catasta sono presenti dei valori salvati. Basterà tener presente che questi non saranno accessibili dall'interno della subroutine chiamata, ma solo dalla routine che li ha salvati. Ad esempio:

```

push bc
call SUB1
pop bc
ret
=====
SUB1 push hl
.
.
pop hl
ret

```

è una struttura perfettamente funzionante, in cui ciascuna routine utilizza solo ciò che ha salvato e libera la catasta dalle "sue cose" prima di ritornare.

Ci resta da esaminare, per quanto riguarda il programma principale, il blocco relativo alla cancellazione del carattere stampato precedentemente.

Il suo funzionamento è molto semplice; dopo aver salvato hl e de, che in quel momento contengono rispettivamente il puntatore ai parametri ancora da recuperare e la distanza fra un parametro e l'altro, esegue una chiamata a subroutine, utilizzando MAIN (cioè il blocco del programma principale) come se fosse un sottoprogramma. Stupiti? In fondo, non cambia niente; l'unica differenza sta nel fatto che il ritorno, al termine di MAIN, avverrà non verso il BASIC, come succede al termine dell'esecuzione normale, ma verso l'operazione seguente alla call.

In questo modo si ha un'esecuzione completa del programma sotto forma di subroutine. I dati con cui avviene questa esecuzione sono però quelli vecchi (è per questo che non bisogna attivare la cancellazione all'atto della prima stampa: non esistono dati precedenti, e si rischia qualcosa di brutto), e quindi si ha una stampa sovrapposta a quella precedentemente eseguita.

Poiché l'immissione dei caratteri nell'area video avviene, per motivi che vedremo, non con

ld <indirizzo> , <linea carattere >

ma con

xor (<indirizzo>), <linea carattere >

la doppia stampa provoca l'annullamento della prima, come abbiamo visto la scorsa volta.

Le subroutine

Ed eccoci alle subroutine.

La prima è PIXEL.

Il suo funzionamento si riduce alla chiamata di un'altra subroutine, presente in ROM all'indirizzo 8881 (22b1 HEX). Questa routine riceve in a la coordinata

RANDOMIZE

y, in c la x, e restituisce in hl l'indirizzo del byte, e in a il bit al suo interno corrispondente alle coordinate date. Come vedete, infatti, prima della sua chiamata avviene un travaso fra b ed a.

La seconda subroutine è DIS, che calcola gli indirizzi successivi.

Si sarebbe potuta senz'altro usare ancora PIXEL, ma il funzionamento di tutto l'insieme sarebbe stato più lento; DIS è infatti notevolmente più corta di quanto non sia la routine in ROM usata da PIXEL.

Il suo funzionamento si basa su alcune semplici considerazioni:

- esistono tre tipi di "salto" fra due byte, l'uno sopra l'altro;
- il primo è quello fra due byte dello stesso carattere (inteso come uno dei 32x22 caratteri normali), ed è di 256;
- il secondo è quello fra l'ultimo byte di un carattere e il primo del carattere sottostante, ed è di -1760;
- il terzo è quello fra l'ultimo byte dell'ultimo carattere di un blocco (uno dei tre blocchi di otto righe ciascuno di cui è formato lo schermo dello Spectrum) e il primo del blocco successivo, ed è di 32.

La distinzione fra questi tre casi avviene in base al valore della coordinata y.

Se il suo valore non è multiplo di otto, si tratterà senz'altro di un altro byte dello stesso carattere, e quindi l'indirizzo contenuto in hl sarà incrementato di 256:

```
inc h
```

Il modo per sapere se il contenuto di a è un multiplo di otto è eseguire

```
and 7
```

Se, infatti, tutti i bit dallo 0 al 2 sono a zero, il valore non potrà che essere 8,16,24,32,...

Se invece il valore della y è un multiplo di otto, siamo di fronte al secondo od al terzo caso.

Prima di distinguere fra i due eseguiamo un'operazione comune ad entrambi, e cioè l'incremento di 32:

```
ld a,l
add a,32
ld l,a
```

infatti,

$$-1760 = -256 * 7 + 32$$

Proprio questa operazione ci dà la chia-

ve per distinguere fra i due casi. Se ci troviamo in presenza del primo byte di un nuovo blocco, l'operazione

```
a+32
```

darà riporto, cioè il suo risultato supererà 255. In questo caso basterà aumentare di 1 h, per obbedire al riporto, ed avremo ottenuto il nuovo indirizzo. Se invece riporto non c'è, occorrerà sottrarre 256*7, cioè eseguire

```
ld a,h
sub 7
ld h,a
```

Riassumendo, la prima scelta avviene sulla base dei multipli di otto, con un salto condizionato verso OUTCHAR.

La seconda scelta avviene invece sulla base della presenza o meno di riporto, con un salto condizionato verso INC-H.

Al ritorno da questa subroutine hl conterrà il nuovo indirizzo.

Ci rimane l'ultima subroutine, PLOT. Qui si inizia col caricare in a il byte (cioè la riga) da stampare. Quindi c viene azzerato, pronto ad accogliere la parte di riga che andrà nel byte successivo (vi ricordate la storia dei due pezzi?).

A questo punto viene effettuato un controllo, per vedere se il contenuto di b (cioè in questo caso il numero del bit di inizio) è zero. A questo servono le operazioni

```
inc b
dec b
```

che influenzano i flag senza utilizzare a, cosa necessaria volendo ottenere lo stesso risultato con una

```
and a
```

Se b è zero, si passa subito alla fase di trasferimento all'area video, non necessitando alcuna suddivisione. Questo perché, in realtà, il dato restituito in a dalla subroutine PIXEL è 7-N, dove N è il numero d'ordine del bit, e quindi un carattere che inizia al bit 7 finisce al bit 0 dello stesso byte.

Lo scorrimento

Nel caso invece si renda necessaria la suddivisione della riga in due parti, questo avviene per mezzo di due operazioni a noi nuove:

```
rra
rr c
```

Si tratta di operazioni di scorrimento. Supponendo che a contenga

11011001

ed il flag di riporto sia a 0, l'esecuzione di

rra

produrrà in a il valore

01101100

e nel flag di riporto il valore 1.

Cosa è successo?

Il contenuto dei vari bit all'interno del registro è "scivolato" verso destra di una posizione. Il posto del bit 7 è stato preso dal contenuto del flag di riporto, mentre quello del bit 0 è andato a finire nello stesso flag.

Questa operazione esiste per tutti i registri, sia verso destra che verso sinistra.

Eseguito in sequenza

rra

rr c

trasferiremo perciò, un bit per volta, il contenuto di a in c. In pratica, avviene un "passamano", in cui a deposita un suo bit nel flag di riporto e c se lo prende.

In realtà anche c deposita un suo bit nel flag, e a se lo prende, ma poiché c è stato a suo tempo azzerato, la cosa non ci dà alcun fastidio.

Questa sequenza di due operazioni viene ripetuta sotto il controllo di b, che contiene appunto il complemento a 7 del numero d'ordine del bit.

RANDOMIZE

Terminata questa fase si passa al trasferimento all'area video.

Questo avviene in due fasi:

xor (hl)

operazione di xor esclusivo del contenuto di a (riga del carattere) e il contenuto della locazione dell'area video in cui andrà inserito. Il risultato di questa operazione è in a.

ld(hl),a

Caricamento del risultato nell'area video.

Perché tutto questo? Non sarebbe più semplice eseguire solo la seconda operazione, cioè il semplice caricamento?

Senz'altro; ma in questo modo si perderebbe quanto c'è già in quella locazione, vale a dire l'eventuale sfondo, od altri caratteri vicini. Visto poi che, come stiamo per vedere, l'operazione di caricamento si estende per due caratteri, il danno sarebbe molto appariscente.

Un altro metodo per inserire senza danni il carattere nel contesto del video potrebbe essere l'operazione di or. In questo caso sarebbe però impossibile la successiva cancellazione, a meno di non distruggere tutto con un

ld (hl),0

Lo stesso procedimento si ripete per il secondo pezzo della riga, che viene inserito nel byte successivo:

ld a,c

inc hl

xor (hl)

ld (hl),a

Per restituire ad hl il suo valore originale l'ultima operazione prima di ritornare è

dec hl

Non prendetevela...

Senza altro questa spiegazione non è stata chiara come avrebbe dovuto. D'altra parte, per esserlo avrebbe riempito non una, ma due o tre riviste.

Dandovi appuntamento alla prossima puntata, in cui vedremo qualcosa di più semplice, ma altrettanto interessante nel funzionamento, invito gli "insoddisfatti" ad esaminare per conto loro, sulla base delle informazioni di questo articolo, la routine presentata. Se le perplessità rimarranno, le loro lettere saranno graditissime.

A presto!



Dalla grande edicola Jackson

Tutto sul personal computer

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana.

9 numeri all'anno: L. 2.000 a numero
Abbonamento: solo L. 15.000

Personal

L'unica rivista indipendente per gli utenti dei personal computer Olivetti.

10 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 35.000

PERSONAL SOFTWARE

Aspetti e problemi del software per personal computer, programmi, giochi e sistemi operativi.

11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 34.000

Bit

La prima rivista europea di personal computer, software e accessori. Con test, novità, analisi del mercato...

11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 43.000

PC

L'unica rivista italiana dedicata ai sistemi MS-DOS, Personal computer IBM e compatibili.

10 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 40.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 Mi

Desidero ricevere GRATIS un numero

della Rivista _____

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

Videocalc

di Paolo Bain

Un piccolo capolavoro di programmazione per lo ZX81!

Questo programma potrà esservi d'aiuto quando vi troverete davanti a tabelle lunghe e complicate di qualsiasi genere: il suo funzionamento ricalca infatti quello dei famosi Visicalc, Multiplan, ecc.

Per quelli che non sanno che cosa sia un tabellone elettronico cercheremo di spiegare brevemente il loro utilizzo.

Facciamo un esempio; dobbiamo calcolare la spesa complessiva della nostra famiglia, perciò immetteremo nel programma le voci più importanti: luce, acqua, gas, ecc., e naturalmente i loro costi, il tutto creando una tabella e ponendo in relazione i dati tra di loro mediante apposite formule.

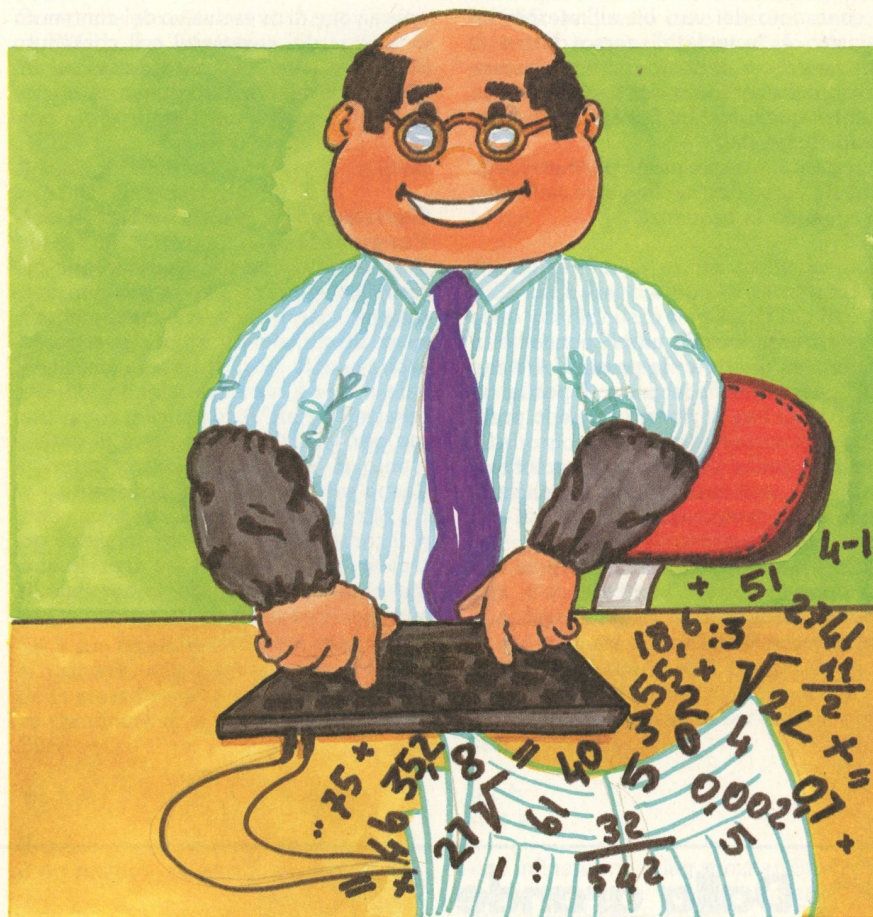
Basterà ora la pressione di un solo tasto per calcolare i totali, naturalmente in base alle formule che avrete prima introdotto; e se un dato cambia, o è destinato a variare nel tempo, voi potrete correggerlo e ricalcolare in un momento il nuovo totale, consentendovi di prevedere cosa accadrà, a seconda della variazione considerata.

I comandi di Videocalc sono parecchi, però prima di elencarvi vorremmo parlarvi del programma in sé.

Questo è molto semplice e si presta facilmente a modifiche: è relativamente corto e possiede molteplici possibilità. C'è anche una funzione di Help, richiamabile in caso di bisogno, che dà brevi spiegazioni di ogni opzione.

Ora torniamo al funzionamento del programma. Dopo averlo caricato con LOAD "VC" o LOAD "", ed aver aspettato circa 3 min., apparirà il nome del programma insieme alla scelta della velocità di esecuzione, FAST o SLOW; questa scelta influirà solo parzialmente, in quanto in certe parti del programma è forzatamente selezionata l'una o l'altra velocità.

Dopo la scelta iniziale compare la lista dei comandi, e premendo ancora un tasto apparirà il tabellone con un grande cursore lampeggiante, in alto a sinistra, che si può muovere con i tasti-freccetta (5,6,7,8). Inoltre, premendo shift+"6"



e shift+"7" il cursore si sposta di due caselle alla volta.

Lo schermo (di 3 caselle per 20) rappresenta solo una parte del tabellone, essendo quest'ultimo infatti lungo in tutto 9 caselle.

Per far entrare nello schermo le parti del tabellone non visibili si usano i comandi Avanti, Indietro e Muovi, vale a dire shift+"A", shift+"I" e shift+"M"; il primo fa scorrere in avanti lo schermo sul tabellone, il secondo lo fa scorrere indietro e il terzo chiede il numero da dove deve cominciare lo schermo.

I comandi relativi alle caselle sono molti; cominciamo con quelli più frequenti: shift+"P" permette di scrivere nelle caselle testi o numeri (in quest'ultimo caso il computer provvederà automatica-

mente ad incolonnarli).

Il comando shift+"0" cancella il contenuto della casella su cui si trova il cursore.

Le funzioni probabilmente più utili sono shift+"1" e shift+"2"; la prima si chiama Memorizza e la seconda Ricopia; si occupano rispettivamente di memorizzare il contenuto numerico di una casella e di riportarlo nella casella occupata dal cursore.

Per quanto riguarda le formule si usa il comando shift+"F", che permette di inserire la formula relativa alla casella dove è posizionato il cursore, indicando una casella come "W(n1,n2)", dove n1 ed n2 sono rispettivamente le righe e le colonne. Se per esempio desideriamo che la casella alla riga A e alla colonna

3 contenga il risultato della somma dei due numeri contenuti nelle caselle situate una all'intersezione di riga 7 e colonna 2 e l'altra all'intersezione di riga B e colonna 1, porteremo il cursore sulla casella in riga A/colonna 3 e, dopo avere digitato shift+"F", inseriremo la formula $W(7,2)+W(B,1)$. Il comando shift+"C" (Calcola) scrive su

tutto lo schermo il risultato di tutte le formule; il comando shift+"T" (Totale), invece, calcola il totale di una formula solo dove è posizionato il cursore. Il comando shift+"W" (Mostra), mostra le formule relative alla casella dove è posizionato il cursore. Inoltre ci sono dei comandi di utilità generale, come shift+"H" (Help), shift +

"K", che visualizza la lista dei comandi ed il loro nome, e shift+"S" per registrare il contenuto del programma. Infine il comando shift+"E" (Esci) permette di uscire dal programma. Una cosa, per finire: nelle formule le coordinate delle caselle possono essere date come appaiono nel lato dello schermo, cioè come lettere.

Videocalc

versione per ZX81 16K

```

000 REM
001 REM
002 REM
003 REM
004 REM
005 SLOW
006 LET L=1
007 LET V$=""
008 LET P$=""
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030 PRINT AT 6,10;"VIDEOCALC";A
T 7,10;"-----";AT 9,13;"BY";
AT 11,10;"BAIN PAOLO";AT 18,2;"U
ELOCITA" (1=SLOW) (2=FAST) ."
031 GOSUB 8604
032 LET V$=INKEY$
033 DIM K$(10)
034 DIM W$(20,9,10)
035 DIM U(20,9)
036 DIM C$(20,9,30)
037 LET X=1
038 LET Y=1
039 LET X1=0
040 LET Y1=0
041 LET A=10
042 LET B=11
043 LET C=12
044 LET D=13
045 LET E=14
046 LET F=15
047 LET G=16
048 LET H=17
049 LET I=18
050 LET J=19
051 LET K=20
052 GOSUB 1600
053 SLOW
054 PRINT AT X,Y;K$
055 LET X=X+X1
056 LET Y=Y+Y1
057 LET K$=W$(X,INT (Y/10+L))
058 PRINT AT X,Y;"
059 LET X1=(2 AND CODE INKEY$=1
13) - (2 AND CODE INKEY$=112) + (INK
EY$="6") - (INKEY$="7")
060 LET Y1=(INKEY$="8") - (INKEY$
="5")
061 IF X1<>0 OR Y1<>0 THEN GOTO
200
062 IF INKEY$=" STOP " THEN GOS
UB 1200
063 IF INKEY$="(" THEN GOSUB 12
50
064 IF INKEY$=">" THEN GOSUB 13
00
065 IF CODE INKEY$=11 THEN GOTO
1345
066 IF CODE INKEY$=119 THEN GOT
O 1400
067 IF INKEY$=" FAST " THEN GOS

```

VIDEOCALC
BY
BAIN PAOLO

```

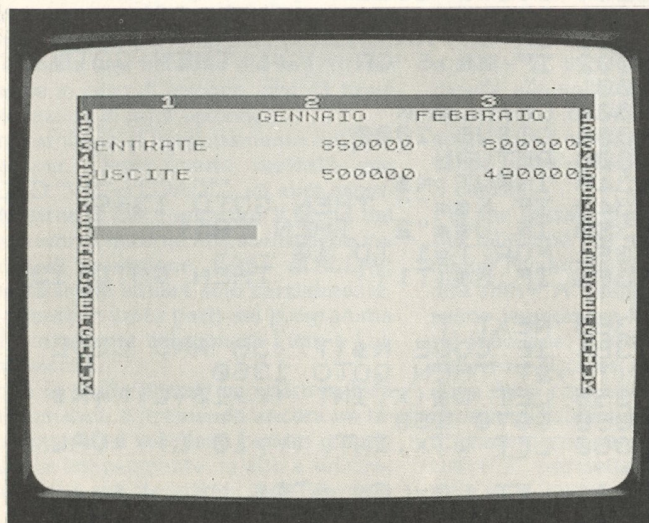
UB 1450
182 IF INKEY$="?" THEN GOSUB 15
00
184 IF INKEY$="<>" THEN GOTO 15
50
186 IF INKEY$="+" THEN GOSUB 16
00
188 IF INKEY$=" LPRINT " THEN G
OTO 2100
190 IF CODE INKEY$=117 THEN GOS
UB 2000
192 IF INKEY$=" AND " THEN GOTO
2050
194 IF INKEY$=" STEP " THEN GOT
O 8500
196 IF INKEY$="*" THEN GOSUB 8
000
198 IF INKEY$=" OR " THEN GOSUB
3000
200 IF X+X1>20 OR X+X1<1 THEN L
ET X1=0
202 LET Y1=Y1*10
204 IF Y1+Y>30 OR Y1+Y<0 THEN L
ET Y1=0
205 GOTO 150
1000 IF V$="2" THEN FAST
1005 PRINT AT 0,0;"
";CHR$ (
156+L);"
";CHR$ (156+L+1)
;"
";CHR$ (156+L+2);"
"
1010 FOR T=1 TO 20
1012 PRINT CHR$ (156+T);W$(T,L);
W$(T,L+1);W$(T,L+2);CHR$ (156+T)
1014 NEXT T
1030 RETURN
1200 IF L>6 THEN RETURN
1202 LET L=L+1
1204 GOSUB 1000
1206 RETURN
1251 IF L=1 THEN RETURN
1252 LET L=L-1
1254 GOSUB 1000
1256 RETURN
1301 INPUT KK
1302 IF KK>6 OR KK<1 THEN GOTO 1
300
1304 LET L=KK
1306 GOSUB 1000
1308 RETURN
1348 INPUT K$
1349 IF K$="" THEN GOTO 1349
1350 IF V$="2" THEN FAST
1351 FOR T=1 TO 10
1352 IF K$(T)<>" " THEN GOTO 135
4
1353 NEXT T
1354 IF CODE K$(T)<38 AND CODE K
$(T)>27 THEN GOTO 1360
1355 LET W$(X,INT (Y/10+L))=K$
1356 GOTO 150
1360 LET W(X,INT (Y/10+L))=VAL K
$
1361 LET LG=LEN STR$ VAL K$
1362 LET W$(X,INT (Y/10+L))=P$(2

```

```

0 TO 29-LG)+STR$ VAL K$
1363 LET K$=W$(X,INT (Y/10+L))
1364 GOTO 150
1400 LET K$=" "
1405 LET W$(X,INT (Y/10+L))=K$
1407 LET W(X,INT (Y/10+L))=0
1410 GOTO 150
1450 IF V$="2" THEN FAST
1451 INPUT H$
1452 IF LEN H$>30 THEN GOTO 1451
1455 LET C$(X,INT (Y/10+L))=H$
1458 RETURN
1500 FAST
1510 FOR T=1 TO 20
1512 FOR R=1 TO 9
1514 IF C$(T,R)<>P$ THEN GOTO 15
30
1516 NEXT R
1518 NEXT T
1520 GOSUB 1000
1522 SLOW
1525 RETURN
1530 LET W(T,R)=VAL C$(T,R)
1533 LET LG=LEN STR$ W(T,R)
1534 LET W$(T,R)=P$(20 TO 29-LG)
+STR$ W(T,R)
1535 GOTO 1516
1550 IF V$="2" THEN FAST
1551 IF C$(X,INT (Y/10+L))=P$ TH
EN GOTO 150
1552 LET K$=STR$ VAL C$(X,INT (Y
/10+L))
1554 LET W$(X,INT (Y/10+L))=K$
1556 LET W(X,INT (Y/10+L))=VAL K
$
1558 GOTO 1351
1600 CLS
1605 PRINT TAB 8;"VIDEOCALC"
1606 PRINT
1608 PRINT "COMANDI : "
1609 PRINT
1610 PRINT "SHIFT+""A"" --> AVA
NTI "
1612 PRINT "SHIFT+""I"" --> IND
IETRO"
1614 PRINT "SHIFT+""0"" --> CAN
CELLA"
1616 PRINT "SHIFT+""T"" --> TOT
ALE"
1618 PRINT "SHIFT+""C"" --> CAL
COLA"
1620 PRINT "SHIFT+""F"" --> FOR
MULE"
1622 PRINT "SHIFT+""M"" --> MUO
VI"

```



```

1624 PRINT "SHIFT+""S"" --> REG
ISTRA"
1626 PRINT "SHIFT+""E"" --> ESC
I"
1628 PRINT "SHIFT+""1"" --> MEM
ORIZZA"
1630 PRINT "SHIFT+""2"" --> RIC
OPIA"
1632 PRINT "SHIFT+""H"" --> HEL
P"
1634 PRINT "SHIFT+""U"" --> MOS
TRA"
1636 PRINT "SHIFT+""P"" --> SCR
IVI"
1700 GOSUB 8600
1705 GOSUB 1000
1707 RETURN
2000 IF V$="2" THEN FAST
2001 LET X$=K$
2002 RETURN
2050 IF V$="2" THEN FAST
2051 LET K$=X$
2052 GOTO 1351
2100 CLS
2102 PRINT "QUALE NOME DAI A QUE
STO LAVORO ?"
2104 INPUT M$
2106 PRINT "QUANDO SEI PRONTO
PREMI UN TASTO"
2108 GOSUB 8604
2110 SAVE M$
2111 LET M$=M$(TO LEN M$-1)+CHR
$(CODE M$(LEN M$)-128)
2112 CLS
2113 PRINT AT 1,8;"BRAIN PAOLO"
2114 PRINT AT 8,(13-LEN M$);"VID
EOCALC :";M$
2116 GOSUB 8600
2120 GOSUB 1000
2125 GOTO 150
3000 IF C$(X,INT (Y/10+L))=P$ TH
EN RETURN
3002 POKE 16418,0
3004 PRINT AT 21,0;C$(X,INT (Y/1
0+L));";";"PREMI UN TASTO ..."
3006 GOSUB 8604
3008 PRINT AT 21,0;"

```

```

3010 POKE 16418,2
3012 RETURN
8000 CLS
8002 PRINT "QUALE COMANDO ?"
8004 INPUT H$
8005 IF H$<>"INDIETRO" AND H$<>"
SCRIVI" AND H$<>"AVANTI" AND H$<>
"COMANDI" AND H$<>"CANCELLA" AN
D H$<>"TOTALE" AND H$<>"FORMULE"
AND H$<>"MOSTRA" AND H$<>"MUOVI
" AND H$<>"REGISTRA" AND H$<>"CA
LCOLA" AND H$<>"ESCI" AND H$<>"M
EMORIZZA" AND H$<>"RICOPIA" THEN
GOTO 8004
8006 PRINT H$;" --> SHIFT+""";
8008 IF H$="SCRIVI" THEN PRINT "
P""";AT 4,0;"PERMETTE DI SCRIVER
E NELLA CASELLA DEI CARATTE
RI O DEI NUMERI."
8010 IF H$="AVANTI" THEN PRINT "
P""";AT 4,0;"FA SCORRERE IN AVAN
TI LO SCHERMO DI 1 CASELLA."
8012 IF H$="INDIETRO" THEN PRINT
"I""";AT 4,0;"FA SCORRERE ALL"
INDIETRO LO SCHERMO DI 1 CAS
ELLA."
8014 IF H$="COMANDI" THEN PRINT
"K""";AT 4,0;"STAMPA TUTTI I COM

```


LOAD

Videocalc

```

ANDI E I          RELATIVI TASTI PER
IL LORO          FUNZIONAMENTO."
8016 IF H$="CANCELLA" THEN PRINT
"0""";AT 4,0;"CANCELLA LA CASEL
LA DOVE STA IL CURSORE."
8018 IF H$="TOTALE" THEN PRINT "
T""";AT 4,0;"STAMPA IL TOTALE DE
LLA CASELLA SECONDO LE FORMULE
INTRODOTTE CON IL RELATIVO COM
ANDO."
8020 IF H$="FORMULE" THEN PRINT
"F""";AT 4,0;"PERMETTE DI INSERI
RE LA FORMULA CHE LEGA 2 O PIU"
CASELLE (RAP-PRESENTATE DA W(N
,N) DOVE N E IL SIMBOLO DELLE LOR
O COORDINATE)."
8022 IF H$="MOSTRA" THEN PRINT "
W";AT 4,0;"PERMETTE DI VEDERE LE
FORMULE RELATIVE ALLA CASELLA
DOVE STA IL CURSORE."
8024 IF H$="MUOVI" THEN PRINT "M
""";AT 4,0;"POSIZIONA LO SCHERMO
PARTENDO DA UN NUMERO INTRODO
TTO PRIMA."
8026 IF H$="REGISTRA" THEN PRINT
"S""";AT 4,0;"PERMETTE DI REGIS
TRARE SU CASSETTA LO SCHER
MO ."
8028 IF H$="CALCOLA" THEN PRINT
"C""";AT 4,0;"CALCOLA I RISULTAT
I SU OGNI CASELLA DEL TABELL
ONE ."
8030 IF H$="ESCI" THEN PRINT "E"
"";AT 4,0;"PERMETTE DI USCIRE DA
L PROGRAMMA"
8032 IF H$="MEMORIZZA" THEN PRIN
T "1""";AT 4,0;"MEMORIZZA IL CON
TENUTO DI UNA CASELLA ( DA USA
RE CON IL COMANDO ""2"" ).
8034 IF H$="RICOPIA" THEN PRINT
"2""";AT 4,0;"RICOPIA NELLA CASE
LLA DOVE C""E"" IL CURSORE, I CAR
ATTERI MEMORIZZATI PREC
EDENTEMENTE CON IL COMANDO "
1""".
8200 GOTO 1700
8500 NEW
8602 PRINT AT 20,4;"PREMI UN TAS
TO ."
8604 IF INKEY$("<>") THEN GOTO 860
4
8606 IF INKEY$="" THEN GOTO 8606
8612 RETURN
9000 IF INKEY$="" THEN GOTO 9000
9002 CLEAR
9004 SAVE "V"
9005 RUN

```

Per ricevere le riviste arretrate di SUPERSINC con cassetta compilate e spedite il coupon qui sotto!

BUONO D'ORDINE PER RIVISTE ARRETRATE
 Anche se l'ordine riguardasse la sola cassetta questa verrà comunque inviata insieme alla rivista al prezzo qui indicato.

Nome
 Cognome
 Via
 CittàC.A.P.Prov.
 Se richiesta fattura:
 Cod. F. e P. Iva
 Data
 Firma
 Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSINC

CEDOLA DI ORDINAZIONE RIVISTE ARRETRATE CON CASSETTA ALLEGATA
 da compilare e spedire in busta chiusa a
 J.soft - Viale Restelli, 5 - 20124 Milano - Tel. 6888228

Vogliate inviarmi i numeri Anno
 di SUPERSINC con cassetta
 Al prezzo di L. 15.000

Contanti allegati Assegno allegato n°
 Ho spedito l'importo a mezzo vaglia postale
 Ho versato l'importo sul cc. n° 19445204 intestato a J. soft - Milano
 Pagherò in contassegno al postino al ricevimento

FINALMENTE!

La Softtrivista che ti gasa!



**QUALCOSA DI SUPER, DI INEDITO,
DI IRRESISTIBILE**

**IL VERO GIOCO
COMINCIA ADESSO**

**IN EDICOLA
JACKSON SOFT
SERIE ORO**

I giochi esclusivi per Commodore 64 e Spectrum 48 K importati dall'Inghilterra, mai presentati in Italia. Una sfida Jackson al già visto, al già fatto, al... già registrato.



continua
la serie delle
entusiasmanti
avventure di
WALLY WEK

Corri in edicola, il vero gioco comincia solo adesso e se sei davvero bravo partecipa alla "sfida al campione", utilizzando il tagliando che troverai sull'ultima pagina di copertina di ogni numero.



**GRUPPO
EDITORIALE
JACKSON**

R*Bert

di **Roberto Corona**

Con questa simpaticissima rivisitazione di un celeberrimo arcade lo Spectrum vi dimostrerà di non essere secondo a nessuno... nemmeno come console per videogiochi!

R*Bert è una versione semplificata del noto gioco da sala Q*Bert, da me realizzata per lo Spectrum 48K (ho cambiato l'iniziale del nome, perché quello originale è marchio registrato della Mylstar Electronics). Per chi non lo conoscesse, ecco di cosa si tratta: il protagonista (R*Bert) saltella su una piramide fatta di cubi; quando saranno stati colorati tutti si passerà al livello successivo. A partire dal quinto livello viene introdotta una difficoltà: non basta saltare una sola volta sui cubi; infatti, al primo salto diventano di un colore intermedio e solo saltandoci sopra un'altra volta diverranno del colore definitivo. Se fosse tutto qui, però, sarebbe troppo semplice e infatti R*Bert ha una nemica: una perfida pallina che piove dall'alto e comincia a saltare anche lei sui cubi, riportandoli al colore di partenza. È ovvio che se R*Bert si lascia raggiungere dalla pallina, perde una vita. Un altro modo di morire è fare un salto fuori dalla piramide di gioco, precipitando nel vuoto (dell'originale ho conservato le simpatiche imprecitazioni che il nostro protagonista proferisce ogni volta che gli capita un incidente), mentre le vite si guadagnano coi punti: una a 100 punti, poi un'altra a ogni raddoppio del punteggio (200, 400, 800, ecc.).

Le apparizioni della pallina sono casuali, ma sono anche legate ai livelli di gioco: più si procede e più spesso verrà, ma soprattutto ai primi livelli i suoi salti sono del tutto casuali, mentre con l'avanzare dei livelli tende sempre più tenacemente ad inseguire R*Bert. Risulteranno in questo caso utili le due "stelle magiche", che dai due cubi estremi della base portano automaticamente sulla cima della piramide (con un decre-



Descrizione delle variabili

a,b	variabili di uso generico in varie parti del programma
c,d,e	usate con a e b per memorizzare gli ATTRibuti da ripristinare nei cubi coperti dal fumetto
i,j	variabili di controllo dei cicli FOR...NEXT
pt	punteggio del gioco
l	vite a disposizione
ll	livello raggiunto
x,y	coordinate di R*Bert
xx,yy	coordinate della pallina
r	numero generato tramite RND per varie scelte casuali
in,in1,in2	colore iniziale, intermedio e finale della faccia superiore dei cubi
att,att1,att2	ATTRibuti relativi a in, in1 e in2
le,ri	colori delle facce laterali dei cubi
z,zz	usate nella lettura della tastiera
ct	conto dei cubi già colorati. Deve arrivare a 21 per completare il livello
ip	incremento da dare a ct quando si colora un cubo. Vale 1 nei primi quattro livelli e 0.5 nei successivi

mento di punti pari al quadrato del livello raggiunto).

Ed ora qualche commento al listato: la prima cosa che salta all'occhio è la sua lunghezza; infatti, nell'intento di aumentarne la velocità di esecuzione (fondamentale in questo genere di programmi) ho deciso di lasciare da parte alcune astuzie di programmazione, che potevano risolvere in poche righe le istruzioni di movimento sia di R*Bert che della pallina, ma, richiedendo un numero più elevato di calcoli, con variabili temporanee, cicli e rimandi, avrebbero reso il gioco meno dinamico. Per lo stesso motivo ho cercato di organizzare tutto il listato in modo che l'esecuzione possa procedere con il minor numero possibile di GO TO e GO SUB.

Ho anche deciso di non ricorrere al sistema diffuso di registrare gli UDG separatamente rispetto al programma, in quanto giudico questo espediente utile soltanto quando consenta il caricamento del programma in 16K, ma nel nostro caso eliminare i DATA non sarebbe sufficiente. L'unico altro motivo che può indurre a registrare i caratteri grafici a parte è che ogni volta che per qualsiasi ragione si ferma il programma, facendolo ripartire con RUN i caratteri vengono nuovamente definiti, e se sono molti occorre attendere qualche secondo, ma ho eliminato questa possibilità con l'istruzione di linea 1150, che controlla se l'ultimo dato dell'ultimo UDG utilizzato è corretto (e quindi se i caratteri grafici sono già stati definiti); in tal caso viene saltato il ciclo di linea 1160. Forse qualcuno avrà già notato la linea 1190 e si domanderà la sua utilità. La variabile "rec", che contiene il record del gioco, viene caricata con un valore che viene letto nella locazione di memoria USR"u" e USR"u"+1, nelle quali all'inizio si trova 100, ma vengono aggiornate con il nuovo valore ogni volta che il record precedente viene superato. Questo fa sì che il valore del record sia memorizzato oltre il RAM-TOP, e non solo nell'area variabili, quindi se il programma viene fermato e fatto ripartire con RUN, tutte le variabili vengono riinizializzate, ma il valore di "rec" non verrà perso. Lo stesso discorso può farsi per la variabile stringa r\$, che contiene il nome del detentore del record e i cui codici dei caratteri si trovano nelle locazioni da USR"u"+2 a USR"u"+12.

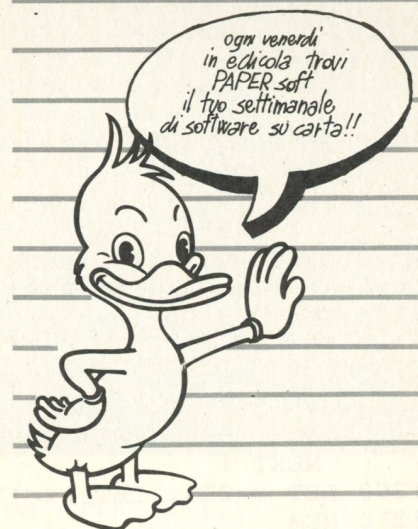
Spero che tutti coloro che amano usare il computer anche per programmare, e non solo per giocare con programmi altrui, possano aver trovato qualcosa di utile in queste mie righe, perché realizzare un programma con le proprie capacità, studiare i possibili miglioramenti, risolvere i problemi più disparati (dall'impaginazione alla velocità di esecuzione) e vedere infine apprezzati i frutti di tale lavoro, oltre ad essere innegabilmente utile per affinare le proprie capacità di programmatore, dà una soddisfazione che nessun giochino altrui potrebbe dare.

v	punteggio da raggiungere per vincere una vita (all'inizio è 100 e raddoppia ogni volta che viene raggiunto)
rec	punteggio record
sg	usata nel volo delle stelle magiche per ricavare da g\$ le giuste coordinate: vale -1 per la stella di destra e 1 per quella di sinistra
s	usata nel suono di Game Over
a\$	porzioni del testo delle istruzioni per ottenere l'effetto tele-scrittore
c\$	10 spazi. È usata per cancellare
f\$	caratteri coperti dall'arrivo della pallina
g\$	valori per le coordinate del volo delle stelle magiche
l\$	usata nell'INPUT del nome di chi ha realizzato il record
r\$	nome dell'attuale detentore del record
t\$	usata per trasferire i caratteri di r\$ nell'area degli UDG

Descrizione del programma

Linee	Commento
1070	la funzione FN a, date le coordinate di schermo x e y, restituisce il relativo indirizzo del File Attributi. Una POKE a tale indirizzo realizza la funzione inversa di ATTR(x,y)
1150	se l'ultimo elemento dei caratteri definibili è quello giusto, significa che il programma ha già girato e vengono quindi saltate le fasi di caricamento UDG e presentazione istruzioni, anche in caso di RUN. Se si vuol cambiare qualche elemento dei DATA, per modificare alcuni caratteri, per ottenerne la lettura e il caricamento battere GO TO 1160. Se si vogliono solo rileggere le istruzioni, scrivere GO TO 3100
1190	decodifica il record e il nome del detentore
1290-1300	la clausola che tra i colori in, in1 e in2 la differenza deve essere almeno 2 (cioè non vi siano colori contigui, come 4-green e 5-cyan, o 0-black e 1-blue) evita che giocando con un TV in bianco e nero ci possano essere dubbi sul fatto che una superficie sia dell'uno o dell'altro colore
1330	il valore di ip è 1 nei primi quattro livelli. Tale valore viene sommato a ct (che è azzerata ad ogni nuovo livello) ogni volta che si colora un cubo, e quando ct arriva a 21 vuol dire che tutti i cubi sono stati colorati. Dal quinto livello ip vale 0.5, in quanto i cubi devono essere colorati due volte ciascuno. Saranno quindi necessari 42 salti per portare ct al valore 21 di fine schema
1520	se non viene premuto nessun tasto, aspetta un tempo inversamente proporzionale al livello raggiunto, prima di passare la mossa alla pallina. Tale tempo va da PAUSE 16 (livello 1) a PAUSE 1 (livello 15 e succ.)
1580	la pausa si ottiene col tasto "P" quando è il proprio turno. Se la mossa spetta alla pallina, questa si muoverà e solo al nostro turno il gioco verrà sospeso.
1620	premando il tasto "G" la pausa viene interrotta e il gioco riprende
2000	si genera un numero casuale, che può andare da 0 a 21 (primo livello) fino a da 0 a 2 (dal 20° livello in poi). Se tale numero è 0, allora entra in campo la pallina
2020-2030	la pallina, entrando in campo dall'alto, copre nella sua discesa una serie di caratteri, che vanno subito rimessi a posto. La successione di questi caratteri è memorizzata in f\$. Se R*Bert si trova sulla sua traiettoria, il carattere corrispondente viene inserito nel corretto punto della stringa f\$
2110	si genera un numero casuale, che al primo livello va da 0 a 4, poi il range aumenta fino al ventesimo livello, in cui può andare da 0 a 21. Se tale numero è minore di 5, il GO TO calcolato porta a una mossa a caso. Se è uguale o maggiore di 5, il GO TO porta all'esecuzione delle linee 2360-2390, che rimandano alla mossa adeguata, a seconda che R*Bert sia più a destra o più a sinistra, più in alto o più in basso della pallina. Con ciò si ottiene che al 1° livello non vi sia alcun inseguimento, ma solo mosse casuali, e col procedere del gioco le mosse casuali diventano sempre più improbabili ed aumentano quelle ad inseguimento

- 2400** capita molto raramente, ma se la pallina con un salto casuale va su una stella magica, morirà tragicamente, regalandoci 10 punti
- 2520** se viene stabilito il record, le due POKE memorizzano il punteggio oltre il RAM-TOP, che quindi non viene azzerato da un eventuale RUN (e neanche dal NEW: se il computer non viene spento, si può ricaricare il gioco e ritrovare il record fatto in precedenza!)
- 2575** premendo il tasto "S", si cambia stato alla variabile logica s, che viene posta a 1 o a 0, attivando o disattivando il suono che accompagna l'invito a giocare ancora
- 2590** scrivendo ripetutamente l'invito a premere il tasto "G", utilizzando l'opzione OVER 1, si ottiene l'effetto di farlo comparire e scomparire.
- La scelta di PAPER 8 (ampiamente sfruttata nel corso del programma) permette di scrivere senza cambiare i colori dei cubi che fanno da sfondo, mentre INK 9 evita che la scritta risulti poco leggibile o addirittura invisibile
- 2630** l'introduzione del nome del giocatore è realizzata tramite INKEY\$, in quanto l'INPUT utilizzerebbe le righe inferiori dello schermo, in cui sono visualizzati i dati relativi al livello e al punteggio raggiunti
- 2640** il CHR\$ 13 corrisponde all'ENTER, e segnala la fine dell'introduzione
- 2650-2660** il CHR\$ 12 corrisponde a DELETE, ed è attivo solo se è presente almeno un carattere
- 2680** per evitare l'accettazione continuata dello stesso carattere, l'elaborazione non va avanti finché il tasto premuto non viene rilasciato
- 2690** se sono stati inseriti 11 caratteri, l'immissione termina senza bisogno di premere l'ENTER
- 2695** i codici dei caratteri che compongono r\$ (il nome del giocatore) vengono posti in coda all'area UDG, come già visto per il punteggio del record
- 2760** la stringa g\$ contiene lo spostamento orizzontale nel volo delle stelle magiche; dalla stella di sinistra tale spostamento è verso destra, e viceversa. Per questo motivo il valore è moltiplicato per sg, che vale 1 o -1, a seconda della stella di partenza
- 2990** questa formula fa sì che il fumetto venga disegnato in ogni caso in una posizione ben determinata, in modo da poter poi ricostruire agevolmente il disegno sottostante
- 3000** prima di disegnare il fumetto si prende nota dei colori (ATTRibuti) di ciò che verrà coperto, che può essere un cubo colorato o da colorare, o una stella o semplicemente lo sfondo
- 3340** l'effetto telescrivente è realizzato da questa linea che analizza la stringa a\$, la quale non deve contenere codici di controllo come quelli dei colori o di INVERSE VIDEO, pena errore K, in quanto tali codici sono formati da due caratteri, mentre con questa istruzione si stampa un solo carattere alla volta



Nella cassetta allegata a questo numero di SUPERSINC troverete un programma omaggio: "Psyco" di Massimo Baldassarre del Gruppo Utilizzatori computer Sinclair e MSX Napoli.

```

1000 REM _____
1010 REM
1020 REM   R*BERT           VI.1984
1030 REM _____ s
      critto da Roberto Corona   via
      Giraldi 8 - 50122 Firenze
      - tel. (055) 262388 -
1040 REM _____
      inizializzazioni generali
1050 CLEAR 65300: POKE 23675,70
1060 BRIGHT 0: FLASH 0: OVER 0:
      INVERSE 0: INK 0: PAPER 0:
      BORDER 0: CLS : RANDOMIZE

```

```

1070 LET r$="": DIM c$(10): DEF FN a(
      x,y)=22528+x*32+y
1080 DATA 0,0,0,0,3,15,63,255,3,15,63
      ,255,255,255,255,255,192,240,252
      ,255,255,255,255,255
1090 DATA 0,0,0,0,192,240,252,255,255
      ,63,15,3,0,0,0,0,255,255,255,255
      ,255,63,15,3
1100 DATA 255,255,255,255,255,252,240
      ,192,255,252,240,192,0,0,0,0,60,
      126,219,255,255,189,195,126
1110 DATA 60,126,255,255,255,255,126,
      60,3,14,28,28,28,28,14,3,192,112
      ,0,0,248,56,56,216
1120 DATA 60,126,219,189,255,255,195,
      126,0,0,0,0,15,63,127,127,0,0,0,
      0,240,252,254,254

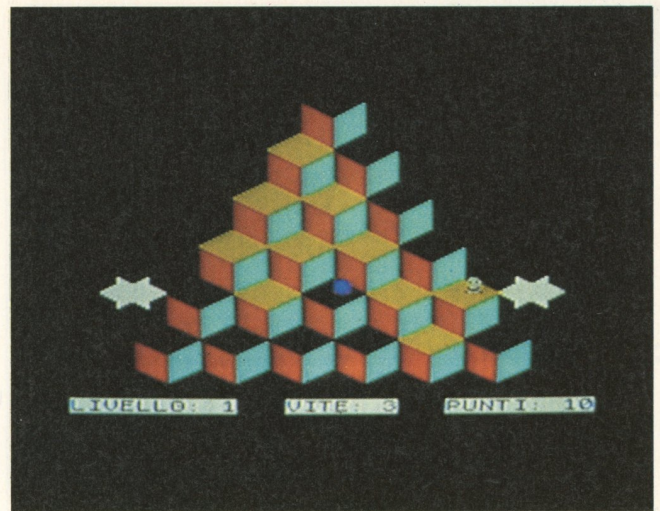
```



```

1700 PRINT AT x,y; PAPER 8;" ":
      POKE FN a(x,y),ATTR (x,y+1):
      LET x=x-1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1710 PRINT AT x,y; INK 8; PAPER 8;"E"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x-1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1720 PRINT AT x,y; INK 8; PAPER 8;"A"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x-1: LET y=y+1: PRINT
      AT x,y; INK 7; PAPER 8;"I"
1730 PRINT AT x,y; PAPER 8;" ":
      POKE FN a(x,y),ATTR (x,y-1):
      LET x=x+1: LET y=y+1: PRINT
      AT x,y; INK 7; PAPER 8;"I"
1740 GO TO 1860
1750 PRINT AT x,y; INK 8; PAPER 8;"C"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x-1: LET y=y-1: PRINT
      AT x,y; INK 7; PAPER 8;"I"
1760 PRINT AT x,y; PAPER 8;" ":
      POKE FN a(x,y),ATTR (x,y-1):
      LET x=x+1: LET y=y-1: PRINT
      AT x,y; INK 7; PAPER 8;"I"
1770 PRINT AT x,y; INK 8; PAPER 8;"A"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x+1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1780 PRINT AT x,y; INK 8; PAPER 8;"E"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x+1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1790 PRINT AT x,y; PAPER 8;" ":
      POKE FN a(x,y),ATTR (x,y+1):
      LET x=x+1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1800 GO TO 1860
1810 PRINT AT x,y; INK 8; PAPER 8;"C"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x-1: LET y=y+1: PRINT
      AT x,y; INK 7; PAPER 8;"I"
1820 PRINT AT x,y; PAPER 8;" ":
      POKE FN a(x,y),ATTR (x,y-1):
      LET x=x+1: LET y=y+1: PRINT
      AT x,y; INK 7; PAPER 8;"I"
1830 PRINT AT x,y; INK 8; PAPER 8;"A"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x+1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1840 PRINT AT x,y; INK 8; PAPER 8;"E"
      : POKE FN a(x,y),ATTR (x,y+1):
      LET x=x+1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1850 PRINT AT x,y; PAPER 8;" ":
      POKE FN a(x,y),ATTR (x,y+1):
      LET x=x+1: PRINT AT x,y; INK 7;
      PAPER 8;"I"
1860 IF ATTR (x+1,y)=71 THEN BEEP 1,
      60: GO TO 2710
1870 IF ATTR (x+1,y)=0 OR x>18 OR (x=
      xx AND y=yy) THEN GO TO 2880
1880 BEEP .04,-10: IF ATTR (x+1,y)=at
      t2 THEN GO TO 1970

```



```

1890 IF ip=1 OR ATTR (x+1,y)=att1
      THEN GO TO 1940
1900 PRINT OVER 1; INK in1; PAPER 8;
      AT x,y-2;" "; INK 7;" "; INK in
      1;" ";AT x+1,y-2;" "
1910 LET pt=pt+1: PRINT #1;AT 1,31-
      LEN STR$ pt; PAPER 7; INK 0;
      BRIGHT 1;" ";pt: IF pt>=v
      THEN BEEP .01,20: BEEP .01,30:
      BEEP .01,40: LET v=v*2: LET l=1
      +(l<9): PRINT #1;AT 1,19;
      PAPER 7; INK 0; BRIGHT 1;l
1930 GO TO 1960
1940 PRINT OVER 1; INK in2; PAPER 8;
      AT x,y-2;" "; INK 7;" "; INK in
      2;" ";AT x+1,y-2;" "
1950 LET pt=pt+1: PRINT #1;AT 1,31-
      LEN STR$ pt; PAPER 7; INK 0;
      BRIGHT 1;" ";pt: IF pt>=v
      THEN BEEP .01,20: BEEP .01,30:
      BEEP .01,40: LET v=v*2: LET l=1
      +(l<9): PRINT #1;AT 1,19;
      PAPER 7; INK 0; BRIGHT 1;l
1960 LET ct=ct+ip: IF ct>=21 THEN
      GO TO 2820
1970 PRINT AT x,y; INK 7; PAPER 8;"I"
1980 REM
      mossa pallina
1990 IF xx>0 THEN GO TO 2100
2000 LET r=INT (RND*(3+(20-11*(11<20)
      ))): IF r>0 THEN GO TO 1510
2010 LET xx=15: LET yy=16
2020 LET f$=" CG AE CG AE J"
2030 IF y=16 AND x<15 THEN LET f$(x+
      1)="I"
2040 FOR i=0 TO 15: LET a=ATTR (i,16)
2050 PRINT AT i,16; INK 1; PAPER 8;"J"
      "
2060 BEEP .001,i+15
2070 PRINT AT i,16;f$(i+1): POKE FN a
      (i,16),a
2080 NEXT i: PRINT AT 15,16; INK 1;
      PAPER 8;"J"

```

```

2090 BEEP .01,15: GO TO 1510
2100 PRINT AT xx,yy; INK 1; PAPER 8;"
      J"
2110 LET r=INT (RND*(4+11-((11-20)*(1
      1>20)))): GO TO 2110+r*10
2120 PRINT AT xx,yy; INK 8; PAPER 8;"
      C": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx-1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2121 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy+1)
      : LET xx=xx-1: PRINT AT xx,yy;
      INK 1; PAPER 8;"J"
2122 PRINT AT xx,yy; INK 8; PAPER 8;"
      E": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx-1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2123 PRINT AT xx,yy; INK 8; PAPER 8;"
      A": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx-1: LET yy=yy-1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2124 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy-1)
      : LET xx=xx+1: LET yy=yy-1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2125 GO TO 2400
2130 PRINT AT xx,yy; INK 8; PAPER 8;"
      C": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx-1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2131 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy+1)
      : LET xx=xx-1: PRINT AT xx,yy;
      INK 1; PAPER 8;"J"
2132 PRINT AT xx,yy; INK 8; PAPER 8;"
      E": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx-1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2133 PRINT AT xx,yy; INK 8; PAPER 8;"
      A": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx-1: LET yy=yy+1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2134 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy-1)
      : LET xx=xx+1: LET yy=yy+1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2135 GO TO 2400
2140 PRINT AT xx,yy; INK 8; PAPER 8;"
      C": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx-1: LET yy=yy-1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2141 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy-1)
      : LET xx=xx+1: LET yy=yy-1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2142 PRINT AT xx,yy; INK 8; PAPER 8;"
      A": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx+1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2143 PRINT AT xx,yy; INK 8; PAPER 8;"
      E": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx+1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2144 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy+1)
      : LET xx=xx+1: PRINT AT xx,yy;
      INK 1; PAPER 8;"J"
2145 GO TO 2400
2150 PRINT AT xx,yy; INK in; PAPER 8;
      "C": POKE FN a(xx,yy),ATTR (xx,y
      y+1): LET xx=xx-1: LET yy=yy+1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2151 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy-1)
      : LET xx=xx+1: LET yy=yy+1:
      PRINT AT xx,yy; INK 1; PAPER 8;
      "J"
2152 PRINT AT xx,yy; INK 8; PAPER 8;"
      A": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx+1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2153 PRINT AT xx,yy; INK 8; PAPER 8;"
      E": POKE FN a(xx,yy),ATTR (xx,yy
      +1): LET xx=xx+1: PRINT AT xx,yy
      ; INK 1; PAPER 8;"J"
2154 PRINT AT xx,yy; PAPER 8;" ":
      POKE FN a(xx,yy),ATTR (xx,yy+1)
      : LET xx=xx+1: PRINT AT xx,yy;
      INK 1; PAPER 8;"J"
2155 GO TO 2400
2360 IF x<xx AND y<yy THEN GO TO 212
      0
2370 IF x<xx THEN GO TO 2130
2380 IF y<yy THEN GO TO 2140
2390 GO TO 2150
2400 IF ATTR (xx+1,yy)=71 THEN FOR i
      =1 TO 5: PRINT AT xx,yy; INK i;
      BRIGHT 1;"S": BEEP .2,60: LET P
      T=PT+2: PRINT #1;AT 1,31-LEN
      STR$ PT; INK 0; PAPER 7;
      BRIGHT 1;" ";pt: NEXT i:
      PRINT AT xx,yy; INK 7; BRIGHT 1
      ;"B": LET xx=0: LET yy=0:
      GO TO 1510
2410 BEEP .01,15: IF xx>18 OR ATTR (x
      x+1,yy)=0 THEN GO TO 2470
2420 IF xx=x AND yy=y THEN GO TO 288
      0
2430 IF ATTR (xx+1,yy)=att THEN
      GO TO 1510
2440 IF ATTR (xx+1,yy)=att1 THEN
      LET ct=ct+.5
2450 PRINT OVER 1; INK in; PAPER 8;
      AT xx,yy-2;" "; INK 1;" ";
      INK in;" ";AT xx+1,yy-2;" "
2460 LET ct=ct-1: GO TO 1510
2470 REM
      morte pallina
2480 PRINT AT xx,yy; PAPER 8;"C":

```



```

IF ATTR (xx+1,yy)=0 THEN LET x
x=xx+1: PRINT AT xx,yy; INK 1;"J
": BEEP .006,28: GO TO 2480
2490 LET xx=0: LET yy=0: GO TO 1510
2500 REM
                fine della partita
2510 LET s=1: PRINT #1;AT 1,19; INK 0
; PAPER 7; BRIGHT 1;"0"
2520 IF pt>rec THEN LET rec=pt:
POKE USR "u",INT (pt/256):
POKE USR "u"+1,pt-256*INT (pt/2
56): GO TO 2600
2530 PRINT AT 4,0; INK 7; BRIGHT 1;"I
L RECORD""APPARTIENE A""
PAPER 4;r$
2540 PRINT AT 4,22; INK 7; BRIGHT 1;"
CON ";rec;AT 5,23;"PUNTI"
2550 FOR i=0 TO 7: FOR j=0 TO 7: IF i
=j THEN GO TO 2580
2560 PRINT AT 0,10; INK j; PAPER i;"
GAME OVER "
2570 IF INKEY$="g" OR INKEY$="G"
THEN BEEP .05,40: BEEP .05,37:
CLS : GO TO 1210
2575 IF INKEY$="s" OR INKEY$="S"
THEN LET s=NOT s: BEEP .2,-38
2580 BEEP .008*s,i*5+j-20: NEXT j
2590 PRINT INK 9; PAPER 8; OVER 1;
AT 17,6;"PREMI KL PER GIOCARE":
NEXT i: GO TO 2550
2600 PRINT AT 0,10; PAPER 5;" GAME O
VER ";AT 4,1; INK 7; PAPER 0;
BRIGHT 1;"HAI FATTO";AT 5,1;"IL
RECORD"
2610 PRINT AT 4,22; INK 7; BRIGHT 1;"
INSERISCI";AT 5,21;"IL TUO NOME"
;AT 6,21; PAPER 4; BRIGHT 0;"
"
2620 LET r$="": LET A=0: PRINT AT 6,2
1; INK 7; PAPER 8;"I";CHR$ 8;
2630 LET l$=INKEY$: IF l$="" THEN
GO TO 2630
2640 IF l$=CHR$ 13 THEN PRINT
PAPER 8;" ": GO TO 2695
2650 IF l$=CHR$ 12 AND a<1 THEN
GO TO 2630
2660 IF l$=CHR$ 12 THEN BEEP .05,0:
PRINT INK 7; PAPER 8;CHR$ 8;"
";CHR$ 8;CHR$ 8;: LET a=a-1:
LET r$=r$( TO a): GO TO 2680
2670 BEEP .05,18: PRINT PAPER 4;L$;:
LET a=a+1: LET r$=r$+l$
2680 IF INKEY$=l$ THEN GO TO 2680
2690 IF a<11 THEN PRINT PAPER 8;
INK 7;"I";CHR$ 8;: GO TO 2630
2695 LET t$=r$+c$+" ": FOR i=1 TO 11:
POKE USR "u"+i+1,CODE t$(i):
NEXT i
2700 GO TO 2550
2710 REM
                volo stelle magiche
2720 LET pt=pt-11*11: PRINT #1;AT 1,3
1-LEN STR$ pt; INK 0; PAPER 7;

```

LOAD

R'Bert

```

BRIGHT 1;" ";pt
2730 LET a=y: LET b=a: LET sg=-1:
IF y<16 THEN LET sg=-sg
2740 PRINT AT 16,a-2;" ";AT 15,a-2
;" "
2750 FOR i=1 TO 14
2760 LET a=a+VAL g$(i)*sg: LET b=b+
VAL g$(i+1)*sg
2770 PRINT AT 17-i,a;" ";AT 16-i,b;
INK 7;"I"
2780 NEXT i
2790 PRINT AT 2,14;" "
2800 LET x=3: LET y=16: PRINT AT x,y;
INK 7; PAPER 8;"I"
2810 GO TO 1860
2820 REM
                livello successivo
2830 LET pt=pt+11*3: IF pt>v THEN
LET l=1+(1<9): LET v=v*2:
BEEP .01,20: BEEP .01,30:
BEEP .01,40: PRINT #1;AT 1,19;
INK 0; PAPER 7; BRIGHT 1;1
2840 PRINT #1;AT 1,31-LEN STR$ pt;
INK 0; PAPER 7; BRIGHT 1;" ";pt
2850 PRINT INK 9; PAPER IN2;AT 0,0;"
COMPLETATO""LIVELLO ";AT 1,8;
11;" BONUS: "" PUNTI";
AT 3,0;11*3
2860 LET 11=11+1: LET ct=0
2870 FOR i=1 TO 20: BEEP .01,1:
NEXT i: GO TO 1270
2880 REM
                morte R*bert
2890 PRINT AT 21,INT (y/4)*4; PAPER 8
;"CDA"
2900 IF x=xx AND y=yy THEN PRINT
AT x,y; INK 7; PAPER 8;"M":
GO TO 3000
2910 FOR i=x+1 TO 21
2920 IF ATTR (x,y)=7 THEN PRINT AT x
,y;" "
2930 LET x=i: BEEP .003,i+22: IF
ATTR (x,y)=0 THEN PRINT AT x,y;
INK 7;"I"
2940 NEXT i
2950 IF ATTR (x,y)=7 THEN PRINT AT x
,y;" "
2960 PRINT #1;AT 0,y; INK 7;"I":
BEEP .003,44
2970 PRINT #1;AT 0,y;" ";AT 1,y;
INK 7;"M": BEEP .003,45
2980 REM
                fumetto
2990 LET y=INT ((y+2)/4)*4

```

```

3000 LET a=ATTR (x-3,y): LET b=ATTR (
x-3,y+3): LET c=ATTR (x-2,y):
LET d=ATTR (x-2,y+3): LET e=
FN a(x-3,y)
3010 PRINT AT x-4,y; INK 7; PAPER 8;"
N{2SG3}O";AT x-3,y; INK 0;
PAPER 7;" S# ";AT x-2,y;" !T ";
AT x-1,y; INK 7; PAPER 8;"PQ
{G3}R"
3020 FOR i=1 TO 30: BEEP RND/10,RND*i
-20: NEXT i
3030 PRINT AT x-4,y; PAPER 8; INK 7;"
";AT x-3,y;"ABCD";AT x-2,y;"
EFGH";AT x-1,y;" ": POKE e,a:
POKE e+1,a: POKE e+2,b: POKE e+
3,b: POKE e+32,c: POKE e+33,c:
POKE e+34,d: POKE e+35,d
3040 IF b=71 THEN PRINT INK 7;
BRIGHT 1;AT x-3,y+1;"CB";AT x-2
,y+1;"GF"
3050 PRINT AT xx,yy; PAPER 8;"C":
POKE FN a(xx,yy),ATTR (xx,yy+1)
3060 IF x<>xx AND y<>yy THEN
LET x=9: LET y=16
3070 PRINT AT xx,yy; INK 8; PAPER 8;"
C": LET xx=0: LET yy=0
3080 LET l=1-1: IF l>0 THEN GO TO 14
90
3090 GO TO 2500
3100 REM
Istruzioni
3110 PAPER 5: INK 0: BORDER 3: CLS
3120 PRINT INK 7; BRIGHT 1;"I";
3130 LET a$=" Questo e' R*bert, il sa
ltatore": GO SUB 3340
3140 PRINT INK 1;"J";
3150 LET a$=" e questa e' la pallina
nemica.": GO SUB 3340
3160 LET a$="Il gioco consiste nel sa
ltellaresui cubi (facendo cosi'
cambiare": GO SUB 3340
3170 LET a$="il loro colore) evitando
di far-si raggiungere dalla pal
lina.": GO SUB 3340
3180 LET a$=" Loschema e' completo q
uando tuttii cubi avranno lo ste
sso colore.": GO SUB 3340
3190 PRINT TAB 5;: LET a$="I tasti da
usare sono:": GO SUB 3340
3195 PRINT AT 12,16; INK 7; BRIGHT 1;
"I"
3200 PLOT 123,84: DRAW -11,11: DRAW 0
,-5: DRAW 0,5: DRAW 5,0: BEEP .0
1,52
3210 PLOT 140,84: DRAW 11,11: DRAW 0,
-5: DRAW 0,5: DRAW -5,0: BEEP .0
1,52
3220 PLOT 123,67: DRAW -11,-11:
DRAW 0,5: DRAW 0,-5: DRAW 5,0:
BEEP .01,52
3230 PLOT 140,67: DRAW 11,-11: DRAW 0
,5: DRAW 0,-5: DRAW -5,0: BEEP .

```

LOAD

R*Bert

```

01,52
3240 INK 2: PRINT AT 10,4;: LET a$="d
a l a 6": GO SUB 3340
3250 PRINT AT 10,21;: LET a$="da `7 a
0": GO SUB 3340
3260 PRINT AT 14,3;: LET a$="da CAPS
S.": GO SUB 3340: PRINT AT 15,6;
: LET a$="a v": GO SUB 3340
3270 PRINT AT 14,22;: LET a$="da B":
GO SUB 3340: PRINT AT 15,22;:
LET a$="a SPACE": GO SUB 3340
3280 INK 0: PRINT ': LET a$="Il tast
o P da' una pausa duranteil gioc
o, mentre durante il ": GO SUB 3
340
3290 LET a$="GAMEOVER col tasto S si
puo'sceglie-re tra suono e silen
zio.": GO SUB 3340
3300 INK 2: PRINT : LET a$="*Buon div
ertimento da R. Corona*":
GO SUB 3340
3310 PRINT #1;AT 0,0; PAPER 3;TAB 31;
" "; PAPER 5;" PREMI KL PER
GIOCARRE "
3320 IF INKEY$<>"g" AND INKEY$<>"
G" THEN GO TO 3320
3330 BEEP .05,40: BEEP .05,37:
PAPER 0: INK 0: BORDER 0: CLS :
GO TO 1180
3340 FOR i=1 TO LEN a$: PRINT "{SG8}"
;CHR$ 8;: BEEP .001,17: PRINT a$
(i);: NEXT i: RETURN
3350 CLEAR : SAVE "R*bert" LINE 1000
5000 REM

```

Programma creato da Rob
erto Corona (055)262388

Reversi

di **Giorgio Simonetti**

SUPERSINC vi propone uno dei più tradizionali giochi su scacchiera

Il gioco del Reversi andava di moda nel '700 ed '800, soprattutto in Inghilterra. È stato chiamato in vari modi, di cui il più famoso è appunto Reversi. Il nome deriva dal fatto che le pedine avevano le due facce di colore diverso e quando una di esse veniva mangiata bastava capovolgerla. Scopo del gioco è impadronirsi di un numero maggiore di caselle della scacchiera rispetto al nostro avversario.

Ci si impadronisce di una casella deponendovi sopra una propria pedina. Ciascun giocatore è contraddistinto da un colore, bianco o magenta; inizia il bianco, si muove a turno in caselle vuote e le prime quattro mosse vanno fatte necessariamente nelle quattro caselle centrali (28,29,36,37), dopo di che valgono le seguenti regole:

- 1) si depone una pedina alla volta e ad ogni mossa bisogna catturare almeno una casella avversaria;
- 2) le caselle avversarie sono catturate se vengono a trovarsi allineate, in orizzontale, verticale o diagonale, con l'ultima pedina deposta ed una precedentemente giocata, senza alcuna casella vuota in mezzo;
- 3) le caselle catturate subiscono il "lavaggio del cervello" e assumono il nostro colore;
- 4) il gioco finisce quando uno dei due giocatori non può più muovere;
- 5) vince chi alla fine del gioco possiede più caselle.

Dopo il messaggio di stop al registratore si possono leggere le istruzioni o giocare direttamente. Due sono le opzioni: due giocatori, con il computer che esegue i controlli e gli spostamenti, oppure



Descrizione del programma

Linee
1

2-10
12-18

28-70
140-184
186-190
195
205-210
335-345

Commento

contiene inizialmente una REM con 40 asterischi, dove il computer pone i risultati delle routine in l/m. Per questo può aversi il caso che, dopo il RUN, con LIST non appaia che la linea 1; per vedere il resto occorre digitare direttamente LIST numero di linea. Può anche accadere che, una volta dato un BREAK, il programma non riparta più istruzioni: m\$ contiene la pagina in stampa preliminari e scelta dell'avversario (se zx=1, si gioca contro il computer)

disegno scacchiera
accettazione e controlli sulla mossa; m\$ contiene la mossa usate in caso di errore (se ok=2)
subroutine per le coordinate di stampa (x e y)
stampano sul video la mossa tramite x e y
cambiano il colore di gioco e aggiornano le variabili: co= colore in gioco, mo= numero della mossa, t= caselle magenta

un giocatore contro il computer. Lo Spectrum provvede a controllare se le mosse sono corrette e chi vince alla fine del gioco. La mossa si esegue digitando il numero della casella che si vuole occupare, seguito da ENTER che rende operativa la mossa. Premendo SPACE prima di ENTER si ha la possibilità di cancellare il numero impostato. Il programma fa uso di alcune routine in linguaggio macchina, che sono collocate nel buffer della stampante. L'area di comunicazione fra il codice macchina e il BASIC è contenuta nella REM della prima linea del programma, che non va tolta né editata. Così com'è, il gioco funziona sul 48K; chi possiede il 16K dovrà digitare il programma, tralasciando le linee relative alle istruzioni.

Note per la digitazione

Prima di tutto battere il listato BASIC e registrarlo su nastro con SAVE "REVERSI" LINE 9000, verificare e dare NEW+ENTER.

Inserire poi il programma "l/m", stando molto attenti a non sbagliare (soprattutto in fase di inserimento dei codici esadecimali). Dare allora RUN e, quando richiesto, registrare il programma di seguito al BASIC. In questo modo il caricamento del linguaggio macchina sarà automatico e il programma partirà da solo. Buon Reversi!!!

410 prime due mosse del computer
3010-3025 messaggio di fine gioco e stampa del vincitore
5500 cambia il colore di gioco
6010-6065 richiamo per le routine in l/m per il controllo di: correttezza della mossa, caselle da mangiare, eventuale mossa del computer
6100 controllo di fine gioco; il gioco finisce se PEEK 2345464
8000-8010 routine di stampa delle istruzioni

Disassemblato del codice macchina

23296	ld a,64	67	ld a,(bc)	20	ret nz
98	ld hl,23472	68	call 23426	21	dec hl
23301	ld (23549),a	71	and a	22	dec d
4	inc hl	72	ld a,(bc)	23	jr nz,23421
5	ld a,l	73	call nz,23384	25	ret
6	cp 241	76	inc bc	26	push hl
8	ret z	77	ld a,(bc)	27	ld l,a
9	ld(23342),hl	78	and a	28	ld a,(23342)
12	call 23340	79	jr nz,23356	31	sub 176
15	ld hl,23761	81	ld(hl),100	33	add a,1
18	ld a,100	83	ret	34	ld l,a
20	ld bc,65	84	ld d,0	35	xor a
23	cpir	86	ld e,a	36	inc a
25	ld a,(23549)	87	ld a,(bc)	37	cp l
28	push ix	88	add a,e	38	jr z,23453
30	pop hl	89	inc d	40	add a,7
31	cp c	90	ld(hl),e	42	cp l
32	jr c,23304	91	inc hl	43	jr z,23453
34	ld a,c	92	ld e,a	45	cp 64
35	ld(23547),hl	93	ld(23400),a	47	jr z,23451
38	jr 23301	96	ld a,3	49	jr 23436
40	ld ix,23536	98	cp(ix-18)	51	pop hl
44	ld hl,23761	23401	jr nz,23412	52	ret
47	xor a	3	ld a,e	53	ld a,(bc)
48	cp(ix+0)	4	call 23426	54	pop hl
51	jr nz,23381	7	and a	55	cp 248
53	ld bc,23789	8	jr z,23421	57	ret z
56	ld a,(bc)	10	jr 23387	58	cp 8
57	ld(23364),a	12	ld a,e	60	ret z
60	ld a,3	13	ld(23419),a	61	xor a
62	cp(ix+9)	16	xor a	62	ret
65	jr nz,23376	17	cp(ix-18)		

Reversi

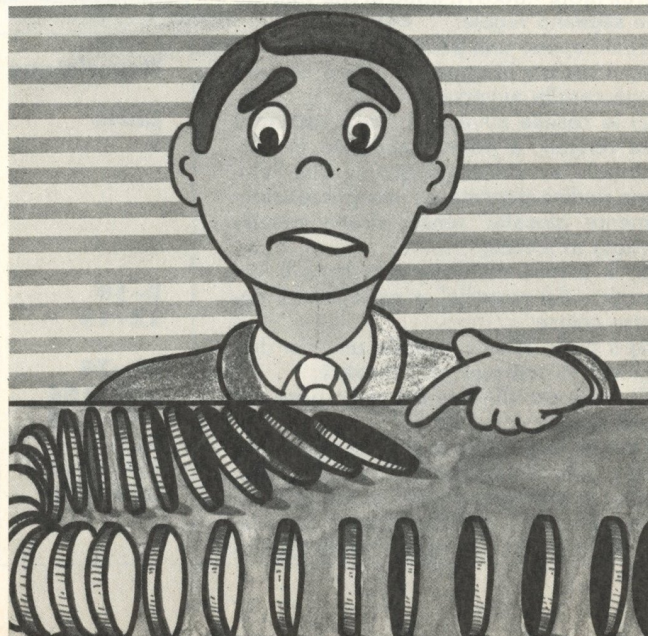
versione per ZX Spectrum 48K
 programma principale

```

1 REM *****
  *****
2 FLASH 1: CLS : FOR v=9 TO 13
  STEP 2: PRINT AT v,3; BRIGHT 1;
  "FERMA IL REGISTRATORE !!!":
  FOR m=0 TO 40 STEP 2: BEEP v*m/
  10000,m: NEXT m: NEXT v
3 FLASH 0: PRINT #0;AT 1,0;" PREM
  I G=Gioco I=Istruzioni ":
  PAUSE 0: GO TO 3+(INKEY$="i")+8
  *(INKEY$="g")
4 PAPER 6: BORDER 6: CLS : PLOT 25
  ,150: DRAW 3,0: PLOT 30,160:
  DRAW -10,0: DRAW 0,-10,-PI:
  DRAW 10,10,-5: DRAW -25,-145:
  DRAW 10,10,5: DRAW 0,-10,PI:
  DRAW 11,0
5 PLOT 20,2: DRAW 210,0: PLOT 20,2
  5: DRAW 210,0: DRAW 0,-23,-2:
  
```

```

PLOT 220,25: DRAW 30,125:
DRAW -8,23,1.3: DRAW -220,0
6 PRINT AT 2,9; INVERSE 1;"REVERSI
  
```



```

": LET m$="Il REVERSI e' un gioc
o dascacchiera per due gioca-tor
i. Regole del gioco: {SG3}Le pr
ime 4 mosse si fan-no solo nelle
 4 caselle centrali (28,29,36,3
7). {SG3}Si muove a turno solo
incaselle libere ed alla condi
zione di allinearsi (anche per d
iagonale) conuna propria pedina
ed in mezzo una o piu' pedine
avversarie senza nessunacasella
vuota interposta "
9 GO SUB 8000: LET m$="{SG3}Le cas
elle avversarie che al momento
della mos-sa si trovano fra le
due amiche cambiano di coloree s
ono quindi catturate. {SG3}Il gi
oco finisce quando tutta la scac
chiera e'oc-cupata o quando uno
dei due giocatori non ha piu'mo
sse possibili. {SG3}Vinc
e chi alla fine del gioco ha piu
' pedine del proprio colore in c
ampo. "
9 GO SUB 8000: LET m$="COME SI GIO
CA {SG3}Le caselle
della scac-chiera sono numerate
da la 64, si muove battendo il
numero della casella che si vuo
le occupare se-guito da ENTER, i
l compu-ter controlla se la moss
a e' corretta ed esegue i necess
ari spostamenti. {SG3}Per corr
eggere premere SPACE prima di E
NTER. {SG3}I colori sono bian
co e magenta,inizia il bianco."
10 GO SUB 8000
12 INPUT ;: PRINT AT 6,0; INVERSE 1
,,,,,
PREMI: " ,,,,,, "
0 = Due giocatori
",,,,,, " 1 = Tu contro il Computer
",,,,,, : PAUSE 0: LET i$=
INKEY$: IF i$<>"0" AND i$<>"
1" THEN GO TO 12
13 LET zx=VAL i$
14 PAPER 7: LET t=0: LET ok=t:
LET m=t: CLS : LET co=INT PI:
LET mo=t: POKE 23624,56:
INPUT ;: FOR v=0 TO 7: POKE
USR "a"+v,128: NEXT v
15 LET m$="-9-8-7-1+1+7+8+9+0":
FOR v=1 TO 9: POKE 23788+v,
VAL m$(2*v-1 TO 2*v): NEXT v
18 FOR v=VAL "23463" TO VAL "23545"
: POKE v,NOT INT PI: NEXT v
28 FOR m=1 TO 27 STEP 3: BEEP .03,m
-30: FOR v=0 TO 21: PRINT AT v,m
;"A": NEXT v: PRINT #0;AT 0,m;"A
";AT 1,m;"A": NEXT m
29 PLOT 8,175: DRAW 192,0
30 FOR v=2 TO 21 STEP 3: BEEP .01,v
: PRINT OVER 1;AT v,1;"

```

```

": NEXT v:
PRINT #0; OVER 1;AT 1,1;"
50 FOR m=1 TO 24 STEP 3: FOR v=1
TO 24 STEP 3: BEEP .0005*v,10+v
+m: PRINT OVER 1;AT m-1,v;(v+2)
/3+(m+2)/3*8-8: NEXT v: NEXT m
65 PRINT AT 0,0;: FOR v=1 TO 22:
PRINT INVERSE 1;" GIOCO DEL *
REVERSI * "(v): NEXT v:
PRINT #0; INVERSE 1;AT 0,0;" ";
AT 1,0;" "
70 BORDER 1: PRINT AT 12,26;"TU ZX
"( TO 6*(zx=1));AT 13,26;
BRIGHT 1; PAPER 7;" ";AT 13,30;
PAPER 3;" ": PLOT 204,44:
DRAW 0,-44: DRAW 48,0: DRAW 0,4
4: DRAW -48,0: PRINT AT 4,26;"GI
OCA";AT 7,26;"MOSSA";AT 9,26;"N"
: CIRCLE 220,103,2
140 GO SUB 300
145 IF zx=0 OR co=3 THEN PRINT AT 1
7,26;" ";AT 20,26;" "
146 IF INKEY$=" " THEN PRINT AT 17,
26;" ";AT 20,26;" "
150 LET i$=INKEY$: IF i$>"9" OR i$<"
10"(m$<>"")+1 TO 1+(m$<>"")
) THEN LET m$="": GO TO 145
155 LET m$m+i$: IF VAL m$m>64
THEN LET m$="": GO TO 145
160 PRINT AT 20,26;"OK ?";AT 17,26;m
$+" ": BEEP .005,40: PAUSE 1:
PAUSE 0: IF INKEY$=CHR$ 13
THEN PRINT AT 20,26;" ":
GO TO 170
166 GO TO 146
174 IF m$=" " THEN LET co=3: GO TO 1
40
175 LET m=VAL m$: GO SUB 195: LET mo
=mo+1: IF (mo<5 AND m<>28
AND m<>29 AND m<>36 AND m
<>37) THEN LET ok=2: GO SUB 1
86: GO TO 150
176 IF mo>4 AND y<24 THEN PRINT
AT y-2,x-1; FLASH 1; BRIGHT 1;"
{SG8}": PAUSE 5+(zx=1 AND co=3)*
15
177 IF mo>4 AND y=24 THEN PRINT #0;
AT 0,x-1; FLASH 1; BRIGHT 1;"
{SG8}": PAUSE 5+(zx=1 AND co=3)*
15
180 IF mo<5 AND PEEK (23472+m)=0
THEN LET t=t+(mo=2 OR mo=4):
GO SUB 200: POKE 23472+m,co:
GO SUB 400: GO TO 140
184 GO SUB 6000: GO TO (140+10*(ok=2
))*(zx=0 OR co=7)+174*(zx=1
AND co=3)
186 IF ok=2 THEN PRINT AT 20,26;
FLASH 1; INK 9; PAPER 2;
BRIGHT 1;"NO!!";AT 17,29;"?":
FOR v=0 TO -20 STEP -1: BEEP .0
06,v: NEXT v: PAUSE 0: LET mo=mo

```

```

-1: PRINT AT 20,26;" ";AT 17,
29;" ": GO SUB 188: IF zx=0
THEN RETURN
187 LET co=3: GO TO 135
188 IF mo>=4 AND y<24 THEN PRINT
AT y-2,x-1; PAPER (PEEK (23472+m
))+7*(PEEK (23472+m)=0);
BRIGHT (PEEK (23472+m)<>0);"
": RETURN
189 IF mo>=4 THEN PRINT #0; PAPER (
PEEK (23472+m))+7*(PEEK (23472+m
)=0); BRIGHT (PEEK (23472+m)
<>0);AT 0,x-1;" "
190 RETURN
195 LET y=3*INT ((m+7)/8): LET x=3*(
m-8*(y/3-1)): RETURN
205 PAUSE 10: POKE 23472+m,co
206 LET y=3*INT ((m+7)/8): LET x=3*(
m-8*(y/3-1))
207 PRINT OVER 1; BRIGHT 1;AT y-3,x
-2; PAPER co;" ": IF y<24
THEN PRINT OVER 1; BRIGHT 1;
AT y-2,x-2; PAPER co;" ";AT y-
1,x-2;" ": RETURN
210 PRINT #0; INK 0; PAPER co;
BRIGHT 1;AT 0,x-2;"A ";AT 1,x-
2;"A ": RETURN
335 BEEP .005,30: PRINT AT 14,26;"
";AT 14,26;mo-t;AT 14,30;" ";
AT 14,30;t: GO SUB 5500: LET m$=
""
340 FOR v=1 TO 3: PRINT AT v,27;
INK co; BRIGHT 1;"{3SG8}":
NEXT v: PRINT AT 9,29;mo
345 RETURN
400 IF zx=0 THEN RETURN
410 LET mo=mo+1: LET co=3: LET t=t+1
: LET m=m+(m=28 OR m=36)-(m=37
OR m=29): PRINT AT 17,26;m;AT 2
0,26; INVERSE 1;" ZX " :
GO SUB 200: RETURN
3010 GO SUB 300: FLASH 1: IF t<>mo-
t THEN PRINT AT 0,25; OVER 1;"
";AT 1,25;" "+"VINCEPERDE"
(1+5*(t>(mo-t)) TO 5+5*(t>(mo-t)
))+ " ";AT 2,25;" IL ";AT 3,25
;" BIANCO";AT 4,25;" "; OVER 0;"
"
3012 IF t=mo-t THEN PRINT AT 1,26;"G
IOCO";AT 2,26;" ";AT 3,26;"
PARI";AT 4,26; FLASH 0;" "
3015 BRIGHT 0: FOR v=10 TO 20 STEP .0
5: BORDER v-10-3*(v>17): BEEP v/
1500,v: NEXT v: PRINT AT 17,26;"
PREMI";AT 18,26;" C ";AT 19,26
;" PER ";AT 20,26;" RUN " :
FLASH 0: PAUSE 0
3020 FLASH 0: IF INKEY$="c" THEN
GO TO 11
3025 GO TO 3020
5500 LET co=co+4-8*(co=7): RETURN
6010 POKE 23361,co+4-8*(co=7): POKE 2
3397,PEEK 23361

```

```

6020 POKE 23342,176+m: RANDOMIZE
USR 23340: FOR v=23761 TO 23791:
IF PEEK v=100 THEN LET ok=2*(v
=23761): LET v=23792: NEXT v:
GO TO 6040
6030 LET m=VAL m$-256*(PEEK v>100)+
PEEK v: GO SUB 195: GO SUB 200
6035 LET t=t+(co=3)-(co=7): NEXT v
6040 IF ok=2 THEN GO TO 186
6050 LET m=VAL m$: LET y=3*INT ((m+7)
/8): GO SUB 195: GO SUB 188:
GO SUB 200: LET t=t+(co=3):
GO SUB 6100: IF zx=0 OR co=3
THEN LET m$="": RETURN
6060 GO SUB 300
6065 POKE 23361,co+4-8*(co=7): POKE 2
3397,PEEK 23361: RANDOMIZE USR 2
3296: LET m$=STR$ (PEEK 23547-17
6): PRINT AT 17,26;m$+" ";AT 2
0,26; INVERSE 1;" ZX ": RETURN
6100 POKE 23361,co: POKE 23397,PEEK 2
3361: RANDOMIZE USR 23296: IF
PEEK 23549=64 THEN GO TO 3000
6110 RETURN
8000 PRINT AT 4,5,: FOR v=1 TO LEN m$
: PRINT m$(v);: BEEP (m$(v)<>"
")*0.005,-5: IF INT (v/25)=v/25
THEN PRINT AT 4+INT (v/25),5-
INT (v/100);: IF v=LEN m$ THEN
PRINT #0;"PREMI UN TASTO PER CO
NTINUARE": PAUSE 0: INPUT ;
8010 NEXT v: LET m$="": FOR v=5 TO 2
STEP -1: FOR m=1 TO 4-(v=2)*2:
PRINT AT 3+m+4*(5-v),v;"
": NEXT m:
NEXT v: RETURN
9000 LOAD ""CODE : RUN

```

Revers 1/m
versione per ZX Spectrum 16K/
48K

```

10 READ m$: LET n=23296: FOR v=1
TO LEN m$ STEP 2: LET a=16*(
CODE m$(v)-48-7*(CODE m$(v)>
VAL "9"))+(CODE m$(v+1)-48-7*(
CODE m$(v+1)>VAL "9")): POKE n,a
: LET n=n+1: NEXT v: SAVE "1m"
CODE 23296,170: VERIFY ""CODE
20 DATA "3E4021B05B32FD5B237DFEF1C8
222E5BCD2C5B21D15C2E64014100EDB1
3AFD5BDDE5E1B938E27922FB5B18D9DD
21F05B21D15CAFDDBE00201C01ED5C0A
32445B3E03DDBE0920090ACD825BA70A
C4585B030AA720E73664C916005F0A83
1473235F32685B3E03DDBE0097BCD
825BA7280B18E77B327B5BAFDDBE00
2B1520FCC9E56F3A2E5BD6B0856FAF3C
BD280DC607BD2808FE40280218F1E1C9
0AE1FEF8C8FE08C8AFC9"

```

Ritorna in edicola

VIDEO BASIC

Il corso più entusiasmante su cassetta
del Gruppo Editoriale Jackson per Commodore 64,
VIC 20 e Spectrum

200.000 copie vendute

del 1° fascicolo della prima edizione

Ogni lezione
uno spettacolo

Con la 1^a lezione
una cassetta giochi
compresa nel prezzo



Il corso è composto da:
20 fascicoli + (Quattordicinali)
20 cassette +
5 splendidi raccoglitori

Oggi è davvero facile imparare il Basic. Con Video Basic il corso su cassetta che ti permette di programmare subito il tuo computer. È facile: tu chiedi, lui risponde, tu impari. Passo dopo passo. Sul tuo schermo appaiono le domande, le risposte, gli esercizi e

tu, senza fatica, presto e bene, impari a conoscere e programmare il tuo computer, sia esso un VIC 20, un Commodore 64 o un Sinclair. Video Basic è in edicola. Provalo subito. Ogni lezione è uno spettacolo.

Oggi il Basic si impara così. Video Basic, il corso su cassetta per parlare subito col tuo computer.

Video Basic
per imparare non solo il Basic.



Un'altra grande idea firmata
GRUPPO EDITORIALE JACKSON

Milano • San Francisco • Londra • Madrid

Quark

di **Luca Ciarlatani**
e **Stefano Di Paolo**

Questo programma vi consentirà di rappresentare sullo Spectrum funzioni matematiche estremamente complesse

Quark è stato realizzato per visualizzare funzioni generiche in tre dimensioni definite tramite due parametri nella forma di:

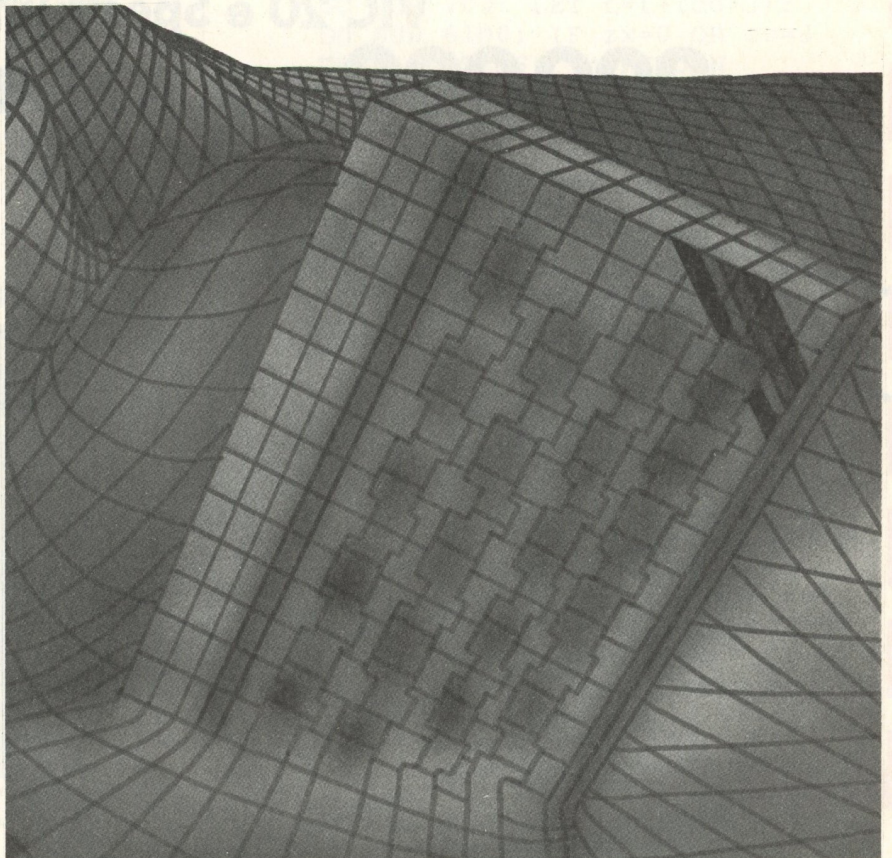
$$\begin{cases} X=f_1(u,v) \\ Y=f_2(u,v) \\ Z=f_3(u,v) \end{cases}$$

I dati necessari per il programma verranno richiesti parte all'inizio e parte durante la sua esecuzione. Tali parametri sono: la funzione stessa, da sostituire a quella presente nelle linee 1020 - 1030 - 1040 (la quale visualizza sullo schermo un toroide, una grossa ciambella); il valore minimo e massimo di u e v; il numero di valori da calcolare per u e per v (chiamati L e M), dal quale dipende il numero di linee e quadrati che verranno rappresentati. L'elaboratore effettuerà la prima fase di calcolo, quindi richiederà altri tre valori (le coordinate cartesiane in tre dimensioni del punto che stabilirà l'angolo della visuale).

Dopo la seconda fase di calcolo sarà richiesto il tipo di visualizzazione preferito. Al termine sarà offerta la possibilità di ripetere il disegno, cambiare angolazione (senza ripetere la prima fase di calcolo), registrare i dati dell'immagine o lo screen\$, cambiare un disegno e cambiare la funzione. È possibile registrare l'immagine video in due modi: o normalmente, tramite uno screen\$, oppure, più rapidamente, registrando in due fasi i dati necessari per il disegno; l'inconveniente di questo secondo tipo di memorizzazione sta nel fatto che il disegno può essere visto solo se caricato con il programma Quark in memoria, anche se in questo modo è possibile alterare i parametri originali della figura.

Questo programma differisce da quelli in commercio in diversi punti:

1) l'esecuzione è più rapida, perché, invece di calcolare tutti i valori assunti dalla funzione in questione, ne calcola tanti quanti l'utente ne desidera, senza che una figura venga compromessa.



Descrizione del programma:

Linee	Commento
10-170	inizializzazione del programma
180-360	calcolo dei valori della funzione
370-590	proiezione, traslazione e rotazione
600-660	trasferimento dei dati necessari alla visualizzazione da un array numerico a uno alfanumerico
670-790	visualizzazione
800-910	opzioni
930-960	subroutine per rototraslazione
970-980	subroutine save disegno
990	save programma con autostart
1000-1070	definizione della funzione

Questo sistema, purtroppo, rende impossibile la cancellazione delle righe nascoste. Anche la rappresentazione grafica è molto rapida, perché tutti i calcoli che necessitano ad essa sono eseguiti prima.

2) Con questo programma è possibile disegnare qualsiasi funzione in tre di-

mensioni, in quanto la funzione (come detto prima) è espressa tramite due parametri e non nella solita forma $Z=f(X,Y)$.

3) C'è la possibilità di visualizzare la funzione sotto tutte le angolazioni senza che sia necessario ricalcolare tutti i valori della funzione (ripetendo, cioè, solo

la seconda parte di rototraslazione, più veloce della prima).

Il programma, come già detto, è diviso sostanzialmente in tre parti:

- calcolo dei valori della funzione;
- proiezione, traslazione e rotazione;
- visualizzazione.

Peculiarità del programma

Il programma calcola i valori delle funzioni partendo non esattamente dai limiti (minimo e massimo di U e V), bensì da poco prima di tale valore, terminando quindi anche prima; ciò non è da considerarsi un errore del programma, ma è stato fatto esclusivamente al fine di evitare che un calcolo vada all'infinito (se ad esempio consideriamo la funzione $\text{SIN}x/x$, normalmente ad x viene attribuito: $-a < x < a$ per ottenere un buon effetto di simmetria, però quando x assume il valore zero si ottiene l'errore "Number too big"; ciò viene così evitato). Tale sistema oltretutto permette di non chiudere la figura, il che consente di vedere una sezione di essa.

Nel caso il programma si arresti dare CONTINUE. Se ciò non fosse sufficiente, per non riscrivere tutti i valori, dare GO TO 80.

Nella fase di inserimento dei limiti di u e v, se non si desidera cambiare il valore, inserire n e il valore non muterà.

Il programma provvede ad ingrandire la figura o a ridurla senza falsare il disegno (è mantenuto il rapporto tra le ascisse e le ordinate).

Durante l'esecuzione del programma si hanno tre fasi di conto alla rovescia.

La velocità di esecuzione della prima fase di calcolo varia a seconda delle difficoltà di esso, mentre la successiva fase ha durata costante, cioè ogni volta i passi di calcolo hanno la medesima durata. È consigliato a coloro che posseggono una stampante di inserire una istruzione di COPY del tipo

```
795 PRINT #1; AT 0,0; "Vuoi stampare? s": PAUSE 0: IF INKEY$="s" THEN COPY
```

La seconda fase impiega 0.4 sec. per passo, mentre la prima e la terza fase sono variabili. In ogni caso la prima fase impiega circa un secondo a passo.

Due cose per finire: 1) "Quark" gira anche sul 16K, ma solo con risoluzioni (valori di L e M) ridotte: agli utenti 16K

converrà modificare il listato con i soliti trucchi «risparmia memoria» (uso massiccio di VAL, sostituzione con una variabile delle costanti numeriche più correnti, riduzione delle linee del pro-

gramma, ecc.) - 2) al termine di "Quark" sulla cassetta troverete un programmino dimostrativo, che caricherà, una dopo l'altra, sei schermate ottenute dagli autori.

Alcune funzioni grafiche interessanti

		valore minimo di	valore massimo di
Sfera:	$X = \text{SIN } V * \text{COS } U$ $Y = \text{SIN } V * \text{SIN } U$ $Z = \text{COS } V$ Utilizzare una buona risoluzione (L = 20 * M = 20)	U = 0 V = 0	U = PI*2 V = PI
Toroide:	$X = (\text{SIN } U + 3) * \text{SIN } V$ $Y = (\text{SIN } U + 3) * \text{COS } V$ $Z = \text{COS } U$	U = 2*PI V = 0	U = 4*PI V = 2*PI
Vulcano:	$X = U * \text{SIN } V$ $Y = U * \text{COS } V$ $Z = (U * U) * \text{EXP}(-U * U)$ Non necessita di alta risoluzione	U = 0.5 V = 0	U = 5 V = 2*PI
Onde:	$X = U$ $Y = V$ $Z = \text{SIN } U * \text{SIN } V$ Per un buon effetto si consiglia di usare alta risoluzione (L = 20 * M = 20)	U = 0 V = 0	U = 2*PI V = 2*PI

Modifica del programma quark:

La parte del programma che calcola i valori della funzione nei vari punti è, come vi sarete sicuramente accorti, molto lenta. Ciò non è dovuto, come in un primo tempo ero portato a pensare, soltanto alla lentezza di calcolo di funzioni complesse come il seno ed il coseno, bensì è causato dall'uso della funzione DEF FN, che rallenta notevolmente l'esecuzione. Quindi suggerisco una modifica:

le linee da modificare sono:

```
1020 DIM C$(3)
1030 LET C$(1) = "fn(x)"
1040 LET C$(2) = "fn(y)"
1050 LET C$(3) = "fn(z)"
280 LET XX = VAL C$(1): (... (il resto rimane inalterato)
290 LET YY = VAL C$(2): (... (il resto rimane inalterato)
300 LET ZZ = VAL C$(3): (... (il resto rimane inalterato)
```

fn(x,y,z) significa ovviamente la funzione di x,y,z.

Nelle linee 280-300 va modificata solo la prima istruzione della linea.

Con questo concludo il materiale inerente il programma Quark.

Quark

versione per Spectrum 16K/48K

```
10 REM Quark
20 GO SUB 1020
```

```
30 INK 7: PAPER 0: BORDER 0:
   BRIGHT 0: FLASH 0: CLS
40 LIST 1020
50 LET t$="
```

```
" : LET t$=t$+t$: LET ui=
0: LET us=ui: LET vi=us: LET vs=
vi: LET l=vs: LET m=1
```

```

60 INPUT "Vuoi cambiare funzione?(s
/n)";a$
70 CLS : IF a$="s" THEN LIST 1020:
PRINT AT 19,0;"Per tornare al p
rogramma :""TAB 21;" RUN + ENTE
R": GO TO 9999
80 LET n=ui: PRINT AT 2,0;"Valore i
nferiore di u:";ui: INPUT ui:
PRINT AT 2,22;ui;t$(22 TO )
90 LET n=us: PRINT AT 3,0;"Valore s
uperiore di u:";us: INPUT us:
PRINT AT 3,22;us;t$(22 TO )
100 INPUT "Vuoi correggere i limiti
di u";a$: IF a$="s" THEN
PRINT AT 2,0;t$: GO TO 080
110 LET n=vi: PRINT AT 5,0;"Valore i
nferiore di v:";vi: INPUT vi:
PRINT AT 5,22;vi;t$(22 TO )
120 LET n=vs: PRINT AT 6,0;"Valore s
uperiore di v:";vs: INPUT vs:
PRINT AT 6,22;vs;t$(22 TO )
130 INPUT "Vuoi correggere i limiti
di v";a$: IF a$="s" THEN
PRINT AT 5,0;t$: GO TO 110
140 LET n=l: PRINT AT 8,0;"Scomposiz
ione di u (L):";l: INPUT l:
PRINT AT 8,23;l;t$(23 TO ):
IF l<=0 THEN GO TO 140
150 LET n=m: PRINT AT 9,0;"Scomposiz
ione di v (M):";m: INPUT m:
PRINT AT 9,23;m;t$(23 TO ):
IF m<=0 THEN GO TO 150
160 INPUT "Vuoi correggere le L e M?
";a$: IF a$="s" THEN PRINT AT 8
,0;t$: GO TO 140
170 INPUT "Vuoi ricorreggere?";a$:
IF a$="s" THEN CLS : GO TO 080
180 CLS : PRINT AT 0,0; FLASH 1;"Att
endere"; FLASH 0""Valore di U:
da ";(STR$ ui+" )( TO 5);" a
";(STR$ us+" )( TO 5)""In "
;l;" passi""Valore di V: da ";
(STR$ vi+" )( TO 5);" a ";(
STR$ vs+" )( TO 5)""In ";m;"
passi"
190 LET passou=(us-ui)/l
200 LET passov=(vs-vi)/m
210 LET v=vs-passov/2
220 LET r=0
230 DIM a(3,1,m)
240 FOR k=m TO 1 STEP -1
250 LET u=us-passou/2
260 FOR j=1 TO 1 STEP -1
270 BEEP .005,20: PRINT AT 18,0;j+(k
-l)*1;" "
280 LET xx=FN x(u,v): LET a(1,j,k)=x
x
290 LET yy=FN y(u,v): LET a(2,j,k)=y
y
300 LET zz=FN z(u,v): LET a(3,j,k)=z
z
310 LET kk=SQR (xx*xx+yy*yy+zz*zz)
320 IF kk>r THEN LET r=kk
330 LET u=u-passou
340 NEXT j
350 LET v=v-passov
360 NEXT k
370 DIM d(3)
380 CLS : BEEP 1,20: INPUT "Inserire
i 3 valori di D (che esp
rimono l'angolo di proie
zione)""D(1):";d(1);TAB 10;"D(2
):";d(2);TAB 20;"D(3):";d(3)
390 IF d(1)*d(1)+d(2)*d(2)<=10e-4
OR d(1)*d(1)+d(2)*d(2)+d(3)*d(3
)>=1000 THEN GO TO 380
400 LET kk=r/SQR (d(1)*d(1)+d(2)*d(2
)+d(3)*d(3))
410 DIM p(3): FOR k=1 TO 3: LET p(k)
=d(k)*kk: NEXT k
420 LET rr=SQR (p(1)*p(1)+p(2)*p(2))
430 LET kk1=-p(2)/rr
440 LET kk12=p(1)/rr
450 LET kk2=-p(1)*p(3)/(r*rr)
460 LET kk22=-p(2)*p(3)/(r*rr)
470 LET kk23=SQR (1-p(3)*p(3)/(r*r))
480 DIM b(2,1,m): LET j=1: LET k=1:
GO SUB 0930
490 LET hormax=b(1,1,1): LET hormin=
hormax: LET vermax=b(2,1,1):
LET vermin=vermax
500 BRIGHT 0: FOR k=m TO 1 STEP -1:
FOR j=1 TO 1 STEP -1
510 GO SUB 0930
520 BEEP .005,20: PRINT AT 18,0;j+(k
-l)*1;" "
530 IF b(1,j,k)>=hormin THEN
GO TO 550
540 LET hormin=b(1,j,k): GO TO 560
550 IF b(1,j,k)>hormax THEN LET hor
max=b(1,j,k)
560 IF b(2,j,k)>=vermin THEN
GO TO 580
570 LET vermin=b(2,j,k): GO TO 590
580 IF b(2,j,k)>vermax THEN LET ver
max=b(2,j,k)
590 NEXT j: NEXT k
600 REM Visualizzazione
610 CLS : LET hor=(hormax-hormin)/25
5: LET ver=(vermax-vermin)/175
620 IF hor<ver THEN LET step=ver:
GO TO 640
630 LET step=hor
640 REM Disegna le righe
650 DIM s$(2,1,m,1): FOR j=1 TO 1:
BEEP .005,20: PRINT AT 18,0;l-j
;" ": FOR k=1 TO m: LET s$(1,j,
k)=CHR$( (b(1,j,k)-hormin)/step)

```

```

: LET s$(2,j,k)=CHR$( (b(2,j,k)-
vermin)/step): NEXT k: NEXT j
660 DIM b(2): LET b(1)=1: LET b(2)=m

670 INK 5: BRIGHT 1: CLS
680 BEEP .1,20: BEEP .1,15: BEEP .3,
20
690 INPUT "Vuoi il reticolato (1) o
i
quadrati (2)?" ;w: IF w
<>1 AND w<>2 THEN GO TO 640

700 CLS : FOR k=1 TO b(2): FOR j=1
TO b(1)-1 STEP w
710 PLOT CODE s$(1,j,k),CODE s$(2,j,
k)
720 DRAW CODE s$(1,j+1,k)-CODE s$(1,
j,k),CODE s$(2,j+1,k)-CODE s$(2,
j,k)
730 NEXT j: NEXT k
740 REM Disegna le colonne
750 FOR j=1 TO b(1): FOR k=1 TO b(2)
-1 STEP w
760 PLOT CODE s$(1,j,k),CODE s$(2,j,
k)
770 DRAW CODE s$(1,j,k+1)-CODE s$(1,
j,k),CODE s$(2,j,k+1)-CODE s$(2,
j,k)
780 NEXT k: NEXT j
790 INK 7
800 PRINT #1;AT 0,0;"Vuoi ripetere i
l disegno? s": PAUSE 0: IF
INKEY$="s" THEN GO TO 670
810 LET e$="
": PRINT #1;AT 0,0;e$
820 PRINT #1;AT 0,0;"Vuoi cambiare a
ngolazione? s": PAUSE 0: IF
INKEY$="s" THEN GO TO 370
830 PRINT #1;AT 0,0;e$
840 PRINT #1;AT 0,0;"Vuoi registrare
l'immagine? s": PAUSE 0: IF
INKEY$="s" THEN GO SUB 0970:
GO TO 800
850 PRINT #1;AT 0,0;e$
860 PRINT #1;AT 0,0;"Vuoi registrare
lo Screen? s": PAUSE 0: IF
INKEY$="s" THEN SAVE "Quark
SCREEN$ "SCREEN$ : GO TO 800
870 PRINT #1;AT 0,0;e$
880 PRINT #1;AT 0,0;"Vuoi caricare u
n disegno? s": PAUSE 0: IF
INKEY$="s" THEN GO TO 0980

890 PRINT #1;AT 0,0;e$
900 PRINT #1;AT 0,0;"Vuoi cambiare l
a funzione? s": PAUSE 0: IF
INKEY$="s" THEN LIST 1020:
PRINT AT 19,0;"Per tornare al p
rogramma:''TAB 20;" RUN + ENTER
": GO TO 9999
910 INK 7: CLS : GO TO 080
920 STOP
930 LET kk=- (a(1,j,k)*p(1)+a(2,j,k)*

```

LOAD

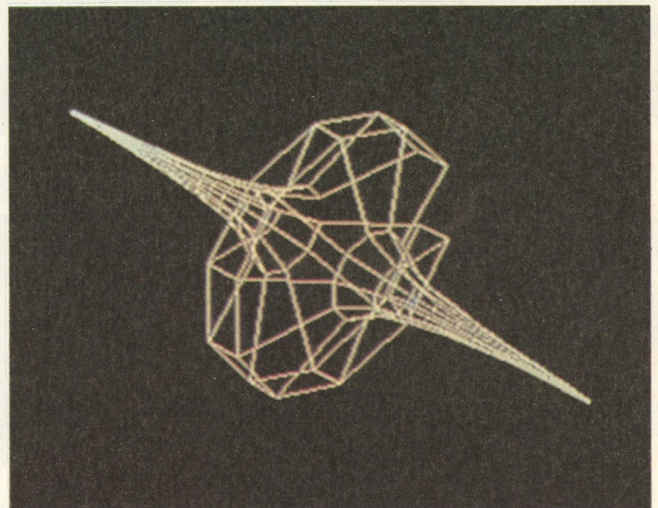
Quark

```

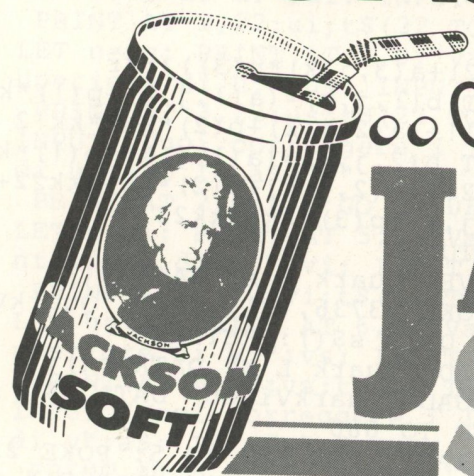
p(2)+a(3,j,k)*p(3))/r*r
940 LET b(1,j,k)=(a(1,j,k)+p(1)*kk)*
kk11+(a(2,j,k)+p(2)*kk)*kk12
950 LET b(2,j,k)=(a(1,j,k)+p(1)*kk)*
kk21+(a(2,j,k)+p(2)*kk)*kk22+(a(
3,j,k)+p(3)*kk)*kk23
960 RETURN
970 SAVE "Quark L M" DATA b():
POKE 23736,181: SAVE "QuarkVide
o" DATA s$(): RETURN
980 LOAD "Quark L M" DATA b():
LOAD "QuarkVideo" DATA s$():
GO TO 680
990 CLEAR : FOR f=1 TO 5: POKE 23736
,181: SAVE CHR$(16+CHR$(5+"-Quar
k-"LINE 0: NEXT f: BEEP .5,20:
RUN

1000
1010
1020 DEF FN x(u,v)=(SIN u+3)*SIN v
1030 DEF FN y(u,v)=(SIN u+3)*COS v
1040 DEF FN z(u,v)=COS u
1070 RETURN

```



SCALDA IL JOYSTICK E GASATI CON



Compilation JACKSON SOFT

**SONO
IN EDICOLA**



**GRUPPO
EDITORIALE
JACKSON**

Milano-Londra-Madrid-San Francisco

IL VERO GIOCO COMINCIA ADESSO

Integrazione numerica

di Giuseppe Voglino

Un programma rivolto a liceali, universitari e a tutti coloro che si dilettono di integrali

La valutazione di un integrale definito $\int_a^b f(x) \cdot dx$ con metodi espliciti è spesso difficile, e qualche volta addirittura impossibile, anche quando la $f(x)$ ha una forma analitica relativamente semplice. In questi casi un'ovvia alternativa è trovare una funzione $g(x)$ che sia una buona approssimazione della funzione $f(x)$ e allo stesso tempo semplice, per poter essere integrata in modo esplicito. La figura 1 illustra l'approssimazione della funzione $f(x)$ con il polinomio $p(x)$, che riproduce esattamente i valori della $f(x)$ nei punti x_0, x_1, x_2, x_3, x_4 .

Il vero valore dell'integrale definito $\int_{x_0}^{x_4} f(x) \cdot dx$ è dato dall'area sottesa dalla curva $f(x)$, laddove l'approssimazione $\int_{x_0}^{x_4} p(x) \cdot dx$ è data dall'area sotto la curva tratteggiata $p(x)$.

Si noti che la differenza tra $f(x)$ e $p(x)$, $\delta(x) = f(x) - p(x)$ differisce in segno sui vari segmenti dell'intervallo di integrazione (nel caso di figura 1 l'intervallo di integrazione è $x_4 - x_0$) e inoltre l'errore totale di integrazione è dato da: $\int_{x_0}^{x_4} \delta(x) \cdot dx = \int_{x_0}^{x_4} f(x) \cdot dx - \int_{x_0}^{x_4} p(x) \cdot dx$, il quale deve essere piccolo, anche se $p(x)$ non è una approssimazione particolarmente buona di $f(x)$. Gli errori positivi in un segmento dell'intervallo di integrazione (vedi $x_0 - x_1, x_2 - x_3$) tendono a cancellare quelli negativi (vedi $x_1 - x_2, x_3 - x_4$) in un altro.

Ci sono molte possibilità di fare ciò, in relazione a: i) scelta della spaziatura dei punti x_i con $i=1, \dots, N$ (dove N è un numero finito); ii) grado del polinomio interpolatore $p(x)$; iii) numero dei punti x_i , ecc..

Nel caso presentato si ricorre all'uso di metodi che valutano la funzione integranda in una decina di punti entro l'intervallo di integrazione. Trattandosi di metodi impiantati su calcolatore, veda il lettore di non includere nell'intervallo di integrazione punti x_i che siano di discontinuità per la funzione, ossia in cui il valore della funzione non sia definito o che valga infinito, e che la funzione

stessa sia continua in tutto l'intervallo di integrazione: insomma, i metodi sono ottimi per tutte quelle funzioni che descrivono fenomeni fisici reali, pena l'overflow della macchina.

I metodi sono i seguenti:

1) SIMPSON

L'intervallo di integrazione è diviso in $2n$ intervalli in cui n è a scelta: aumentando n (di solito $n=100$ è più che sufficiente) aumenta il tempo richiesto dal calcolatore per eseguire l'integrale.

La formula da impiegare è la seguente:

$$\int_a^b f(x) \cdot dx = \frac{h}{3} \cdot [f_0 + 4 \cdot (f_1 + f_3 + \dots + f_{2n-1}) + 2 \cdot (f_2 + f_4 + \dots + f_{2n-2}) + f_{2n}]$$

avendo denotato:

$$h = \frac{b-a}{2n}; f_i = f(x_i) \text{ e } x_i = a + i \cdot h$$

2) BODE

La formula usata è la seguente:

$$\int_a^b f(x) \cdot dx = \frac{5 \cdot h}{299376} \cdot$$

$$\cdot [16067 \cdot (f_0 + f_{10}) + 106300 \cdot (f_1 + f_9) - 48525 \cdot (f_2 + f_8) + 272400 \cdot (f_3 + f_7) - 260550 \cdot (f_4 + f_6) + 213684 \cdot (f_5 + f_5)]$$

$$\text{dove: } h = \frac{b-a}{10}; f_i = f(x_i)$$

$$\text{e } x_i = a + i \cdot h \text{ } 2/3 \text{ sp. per } i=0, 1, \dots, 10$$

3) GAUSS

La formula impiegata è la seguente:

$$\int_a^b f(x) \cdot dx = \frac{b-a}{2} \cdot \sum_{i=0}^{11} W_i \cdot f(y_i)$$

$$\text{dove } y_i = \frac{(b-a)}{2} \cdot x_i + \frac{(b+a)}{2}$$

e inoltre:

$$W_0 = W_6 = 0.2491470458$$

$$W_1 = W_7 = 0.2334925365$$



$$\begin{aligned} W_2 = W_8 &= 0.2031674267 \\ W_3 = W_9 &= 0.1600783285 \\ W_4 = W_{10} &= 0.1069393259 \\ W_5 = W_{11} &= 0.0471753363 \end{aligned}$$

$$\begin{aligned} x_0 = -x_6 &= 0.1252334085 \\ x_1 = -x_7 &= 0.3678314989 \\ x_2 = -x_8 &= 0.5873179542 \\ x_3 = -x_9 &= 0.7699026741 \\ x_4 = -x_{10} &= 0.9041172563 \\ x_5 = -x_{11} &= 0.9815606342 \end{aligned}$$

4) CHEBYSHEV

In questo caso si usa la seguente formula:

$$\int_a^b f(x) \cdot dx = \frac{b-a}{9}$$

$$\cdot \sum_{i=0}^8 f\left(\frac{b-a}{2} \cdot x_i + \frac{b+a}{2}\right)$$

in cui:

$$\begin{aligned} x_0 &= 0 \\ x_1 = -x_5 &= 0.1679061842 \\ x_2 = -x_6 &= 0.5287617831 \\ x_3 = -x_7 &= 0.6010186554 \\ x_4 = -x_8 &= 0.9115893077 \end{aligned}$$

Nel caso in cui il secondo estremo di integrazione è infinito (anche qui prestare molta attenzione alla funzione da integrare, altrimenti si incorre in un overflow del computer oppure in un risultato tutt'altro che attendibile) si fa uso della seguente:

$$\int_a^{+\infty} f(x) \cdot dx = 2 \cdot \sum_{i=0}^{11} \frac{W_i}{(1+x_i)^2}$$

$$\cdot f\left(\frac{2}{1+x_i} + a - 1\right)$$

e i coefficienti sono quelli di Gauss.

L'intero programma comprende un sottoprogramma principale, il quale si occuperà dell'introduzione dei dati in ingresso, della stampa sul video dei risultati ottenuti e delle varie opzioni contenute nel programma stesso, nonché richiamerà il sottoprogramma voluto dal tipo di integrazione scelto (Simpson, Bode, Gauss o Chebyshev).

Il diagramma di flusso del programma principale è rappresentato in figura 2.

Sottoprogramma SIMPSON (linee 100÷150)

Il primo ciclo FOR...NEXT (linea 110) calcola la quantità $(f_1 + f_3 + \dots + f_{2n-1})$, dove la quantità $2n-1$ è stata chiamata $n1$, relativa alla somma dei valori della funzione integranda calcolata nei punti dell'intervallo (a, b) di posto dispari: il valore di tale somma viene posto nella variabile st di cui alla linea 110; quindi la prima volta la variabile di controllo i del ciclo della linea 110 parte da 1 e arriva al valore $n1$, incrementandosi di 2 unità ad ogni passaggio: ciò si fa age-

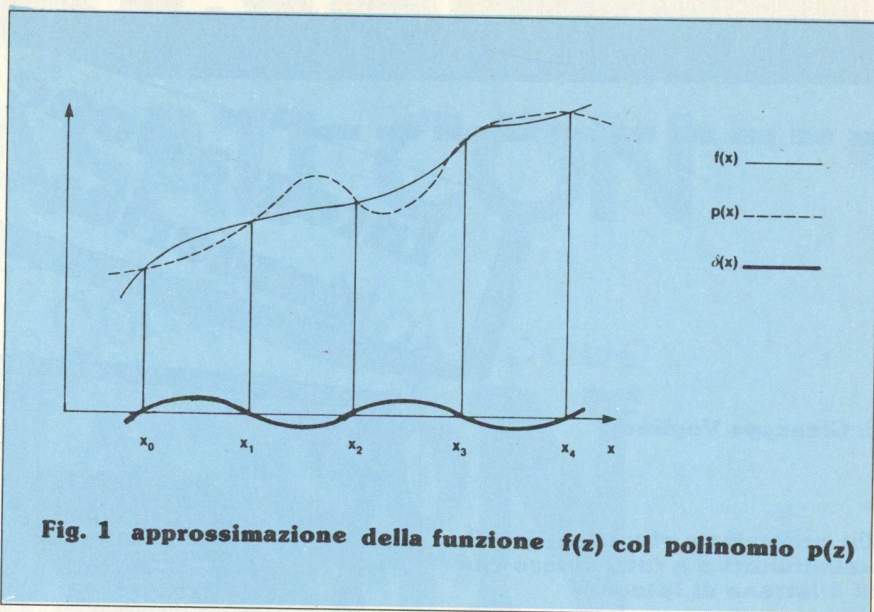


Fig. 1 approssimazione della funzione $f(z)$ col polinomio $p(z)$

volmente con FOR i=1 TO n1 STEP 2. Salvato il valore calcolato st , e moltiplicato per 4 nella variabile nt , si utilizza ancora la variabile st , dopo averla posta a zero, per il calcolo della somma della funzione integranda nei punti del medesimo intervallo di posto pari, ossia di $(f_2 + f_4 + \dots + f_{2n-2})$, dove la quantità $2n-2$ è stata chiamata $n2$; in tal caso la variabile di controllo del secondo ciclo parte da 2 ed arriva a $n2$: FOR i=2 TO n2 STEP 2.

Alla linea 140 il valore di st , calcolato nel secondo ciclo, viene sommato alla quantità contenuta in nt e ai valori della funzione calcolata negli estremi dell'intervallo, che sono a e b . Il tutto viene poi diviso per 3 e moltiplicato per $h = (b-a)/2n$.

Sottoprogramma BODE (linee 200÷250)

Tale metodo valuta la $f(x)$ in 12 punti, i quali a coppie, e precisamente il primo e l'ultimo, il secondo e il penultimo, ecc.; vengono moltiplicati per dei coefficienti determinati a priori.

Un primo approccio consisterebbe nel solito ciclo FOR...NEXT e il contatore i variabili da 0 a 10, ma ci si rende conto che i coefficienti sono solo 6 e che quando i è maggiore o uguale a 5 i coefficienti si ripetono. Allora, posti i coefficienti in un vettore $b(s)$ di lunghezza pari a 6, si utilizza un ciclo FOR...NEXT in cui $i=0, 1, \dots, 5$; si noti poi che quando si calcola la funzione nel primo punto, vedi f_0 , e i è ora pari a zero, occorre calcolare anche la funzione nell'ultimo punto, vedi f_{10} , poiché entrambi vanno moltiplicati per il medesimo coefficiente, che in questo caso è $b(1)$ e dato che $i=0$ risulta $b(1) = b(i+1)$: ossia bisognerà fare un'operazione tipo $b(i+1) (f_i + f_j)$ per $i=0, \dots, 5$ in cui $j=10-i$.

FOR i=0 TO 5: LET $x=a+ih$: LET $st=FN f(x)$: LET $j=10-i$: LET $x=a+jh$: LET $st=st+FN f(x)$: LET $nt=nt+b(i+1) st$: NEXT i

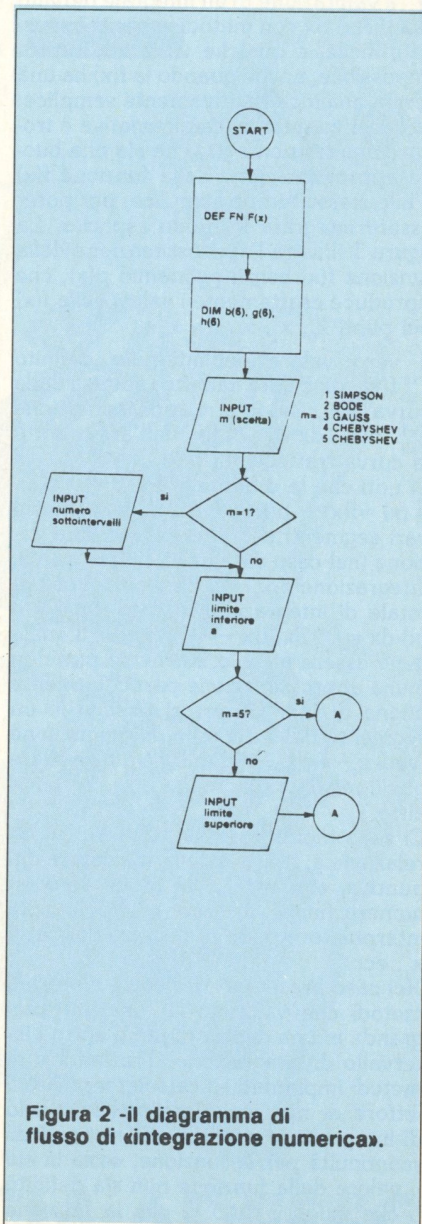


Figura 2 -il diagramma di flusso di 'integrazione numerica'.

Sottoprogramma GAUSS (linee 300 ÷ 340)

Si potrebbe tradurre direttamente in BASIC il diagramma di flusso presentato, ma ci si accorge che i 12 coefficienti $W(s)$ non sono tutti diversi, ma sono uguali a coppie (W_0, W_6) , (W_1, W_7) , ... (W_5, W_{11}) e anche gli $x(s)$ in cui risulta essere $(x_0, -x_6)$, $(x_1, -x_7)$, ..., $(x_5, -x_{11})$. Quindi si memorizzano i $W(s)$ in un vettore $g(s)$ di lunghezza pari a 6, che conterrà W_0, \dots, W_5 , visto che da W_6 a W_{11} si ripetono uguali, e gli $x(s)$ in un altro vettore $h(s)$, sempre di lunghezza 6, che conterrà x_0, \dots, x_5 , dato che da x_6 a x_{11} cambia solo il segno.

Posto $h=(b-a)/2$, si calcola $x=((b-a)/2) \cdot g(i) + (b+a)/2$ in due volte: prima $x = \text{sign} \cdot g(i) \cdot h$ in cui $\text{sign} = 1$ e, osservando che $(b+a)/2 = (b-a)/2 + a = h+a$, si ha poi $x = x+h+a$; in seguito è $nt = nt + h(i) \cdot \text{FN } f(x)$. Tutto questo in un ciclo FOR... NEXT per $i=1, \dots, 6$. Uscito dal ciclo si cambia $\text{sign}=1$ in $\text{sign}=-1$, in modo da cambiare segno ai $g(i)$ e si ripete il ciclo interno in i un'altra volta.

Sottoprogramma CHEBYSHEV (linee 400÷470)

Identico ragionamento anche in questo metodo, con l'unica differenza che il ciclo più interno ha $i=1, \dots, 4$ anziché $i=0, \dots, 5$, dato che qui i coefficienti sono 4 e che $x_0=0$

Sottoprogramma CHEBYSHEV con estremo superiore infinito (linee 500÷550)

Il metodo n° 5 ricopia nella sequenza delle istruzioni quello di GAUSS e di questo utilizza i medesimi coefficienti. Tuttavia si è preferito spezzettare la formula in operazioni più semplici, cosa che permette al calcolatore di diminuire il tempo di calcolo della stessa: per esempio, porre la quantità $1+x(i)$ in una variabile di comodo, vedi q, e quindi non dover più ripetere l'operazione $1+x(i)$ nel passo successivo; e anche evitare la $(1+x(i))^2$, che viene sostituita da q^*q , cosicché nel caso in cui la quantità $1+x(i)$ dovesse risultare negativa si evita un arresto del programma, come risulterebbe se si scrivesse $(1+x(i))^2$. Nel caso si scelga il metodo n° 5 bisogna porre naturalmente attenzione nella funzione da integrare (si rimanda la questione a testi specializzati sulla materia); degli esempi di guida: evitare $f(x) = \cos(x)$, $\sin(x)$, x , x^2 , x^3 , ..., x^n , ecc...

Programma principale

Caricato il programma appariranno sullo schermo le istruzioni per poter inserire la funzione da integrare: niente di più semplice. Occorre solo digitare una semplice linea di programma 10 DEF FN f(x)= ..., dove al posto dei puntini vi sarà la $f(x)$ scritta in modo esplicito e il numero di linea 10 è tassativo, visto che tale linea deve essere inserita proprio in un punto preciso del program-

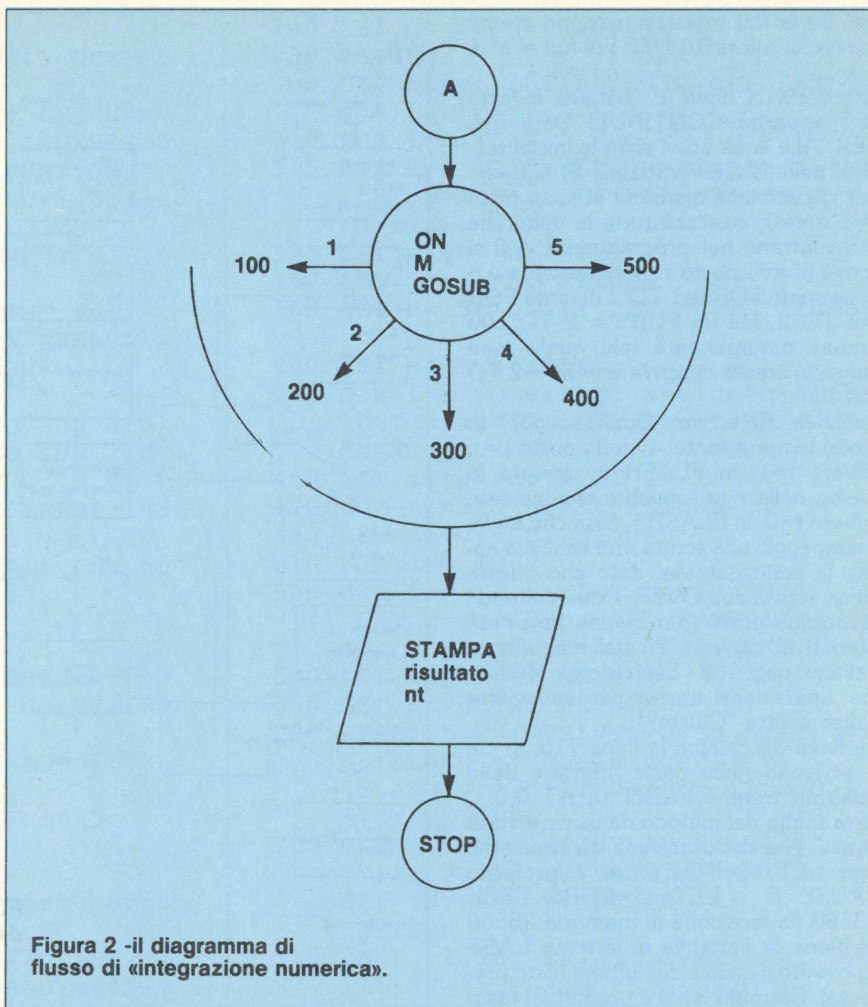


Figura 2 - il diagramma di flusso di «integrazione numerica».

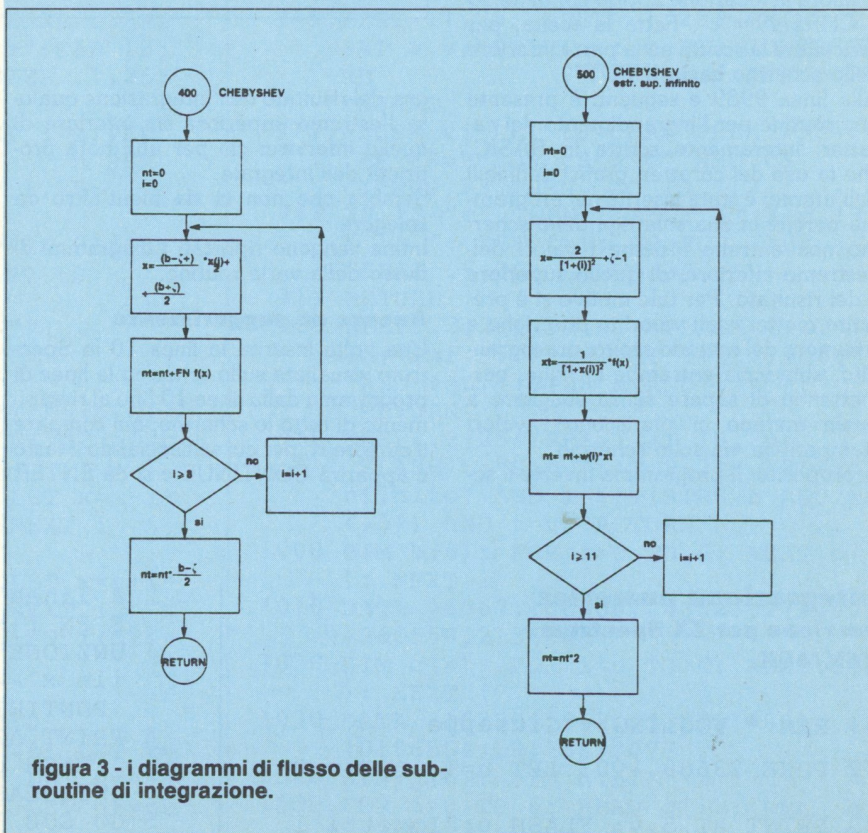


figura 3 - i diagrammi di flusso delle sub-routine di integrazione.

ma. Se la $f(x)$ fosse l'equazione di una parabola, allora $10 \text{ DEF FN } f(x) = x^2 + x + 1$.

Introdotta la linea 10 battere il tasto "c": apparirà CONTINUE. Date ENTER. Alla linea 15 vi sono le inizializzazioni delle diverse variabili. Si noti che per risparmiare memoria si sono posti $u=1$ e $v=0$, cosicché tutte le volte che si incontrano nel programma 1 o 0 si scrive al loro posto rispettivamente u o v; esempio FOR $i=1$ TO 2 diventa FOR $i=u$ TO 2, ma un FOR $i=2$ TO 100 rimane naturalmente tale quale (non venga in mente di scrivere FOR $i=2$ TO uvv !!!).

La linea 30 scrive "Quale scegli?" in modo lampeggiante, avendo posto $t=u$, ovvero $t=1$, in FLASH t; operata la scelta, nella riga seguente si pone $t=v$, ovvero $t=0$, in FLASH t, cosa che toglie il lampeggio alla scritta, ma cancella anche la scritta stessa, dato che questa viene scritta con OVER 1 due volte nel medesimo posto (per questa proprietà riferirsi al capitolo 25 del manuale in italiano, pag. 168 - Esercizio 1). Medesima operazione anche per cancellare l'altra scritta "Giusto?".

La linea 36 chiama la linea 715, in cui si scrivono nella parte inferiore dello schermo tramite PRINT 0; AT 0,0:... se la scelta del metodo da usare è stata giusta. Anziché usare una istruzione del tipo LET $t\$=INKEY\$$ si è preferito PAUSE 0 : LET $t\$=CHR\$ PEEK 23560$ (la locazione di memoria 23560 contiene la variabile di sistema LAST K, ovvero il codice dell'ultimo tasto premuto). Esempio: se si è premuto il tasto c, allora la locazione 23560 contiene 99 e $CHR\$(99)="c"$. Fatta la scelta, per cancellare la scritta nella parte inferiore dello schermo basta INPUT "".

Alla linea 9989 e seguenti è presente una routine per l'ingrandimento dei caratteri interamente scritta in BASIC, che fa uso dei caratteri grafici definibili dall'utente; è stata inserita nel programma perché in una sola riga dello schermo non entrano insieme i valori dell'estremo inferiore, di quello superiore e del risultato. Per tale motivo si è preferito mettere tali valori in due righe e il numero del metodo scelto, ora ingrandito, abbraccia entrambe le righe, permettendo di sapere senza sbagliare a quale metodo si riferiscono i valori stampati via via sullo schermo.

A proposito: il programma inverte il se-

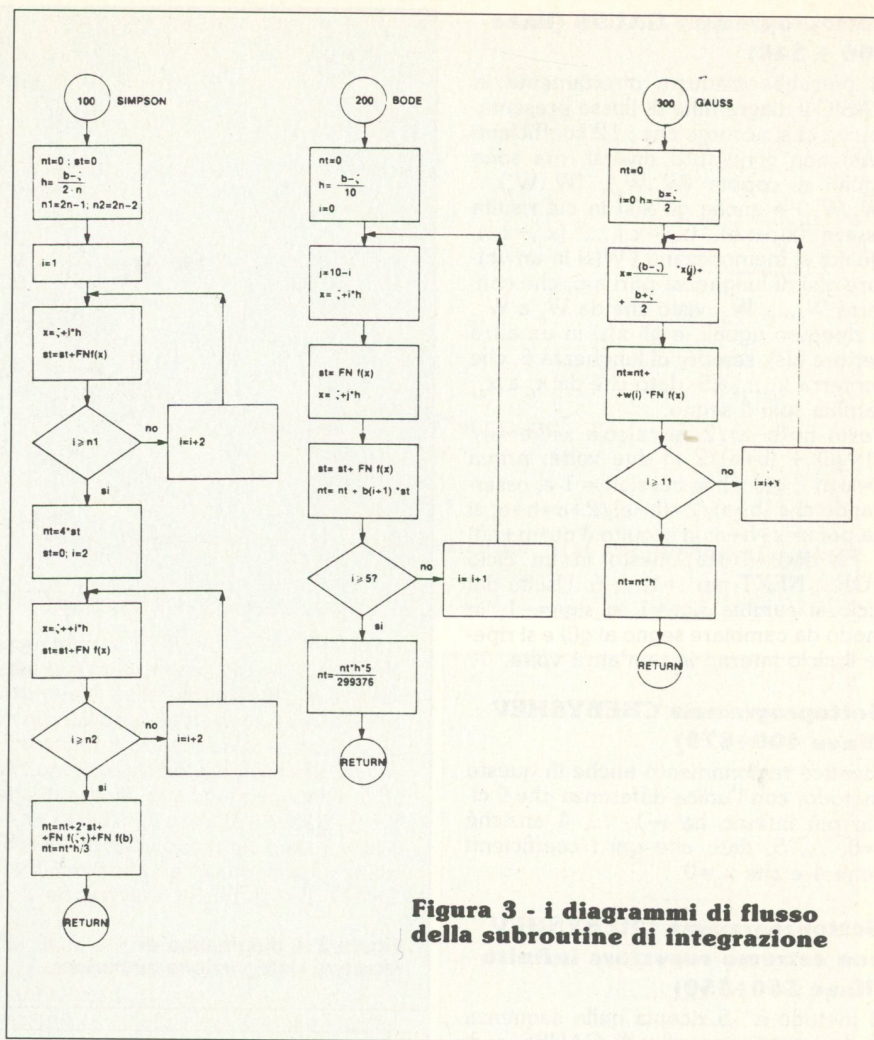


Figura 3 - i diagrammi di flusso della subroutine di integrazione

gno del risultato dell'integrazione qualora l'estremo superiore sia inferiore di quello inferiore: ciò per una nota proprietà dell'integrale.

Sembra che non ci sia nient'altro da spiegare.

Infine vengono riportati i diagrammi di flusso delle varie routine.

Ancora un suggerimento

Una volta inserita la linea 10 lo Spectrum visualizza sullo schermo le linee di programma dalla linea 10 fino al riempimento di tutto lo schermo; qui compare il cursore K, per cui schiacciando il tasto c apparirà CONTINUE e si dà ENTER

per la ripresa del programma.

Se si vuole evitare la visualizzazione delle righe di programma, è sufficiente editare la linea 1 in questo modo:

```
1 REM PREMERE CONTINUE
```

e, prima di dare ENTER, entrare in modo E premendo CAPS SHIFT + SYMBOL SHIFT, poi premere CAPS SHIFT + 7, cosicché le linee seguenti verranno scritte in bianco e scompariranno dal video, e infine dare ENTER. In questo modo le linee di programma successive alla linea 1 non verranno visualizzate sullo schermo e comparirà in alto solo la linea 1, che indicherà che cosa bisognerà fare.

Integrazione numerica versione per ZX Spectrum 16K/48K

```
1 REM © VOGLINO Giuseppe
2 POKE 23609,100: LET u=1: LET v=0
3 PRINT AT 5,v; FLASH u;"Immetti l
```

```
a linea"; FLASH v;AT 8,5;" 10 DE
F FN f(x)=...";AT 13,8;"CON LA F
UNZIONE""TAB 5;"-- nella variab
ile x --""TAB 6;" AL POSTO DEI
PUNTINI"
```

```
4 PRINT AT 20,v;"POI PREMERE ";
FLASH u;"CONT"; FLASH v;" PER C
ONTINUARE"
5 GO SUB 1000: BEEP 1,25: STOP
```



```

6 DIM n$(5,9)
7 FOR i=u TO 5: READ n$(i): NEXT i

15 CLS : LET l=v: LET a=v: LET b=v
20 PRINT AT v,u; FLASH u; INVERSE u
;"INTEGRAZIONE DI UNA FUNZIONE"
25 FOR i=u TO 5: PRINT AT 2+i,0;i;
TAB 3;n$(i): NEXT i
30 LET t=u: GO SUB 700: INPUT m
31 LET t=v: GO SUB 700
33 IF m<l OR m>5 THEN PRINT #v;
AT v,v; FLASH u;"SCELTA ERRATA -
Ripeti, prego!"; FLASH v:
PAUSE 200: GO TO 30
35 GO SUB 705: LET t=u: GO SUB 710
36 GO SUB 715: LET t=v: GO SUB 710
37 PRINT AT 10,v; INVERSE u;"N.";
INVERSE v;AT 9,5;"- ESTREMI INT
EGRAZIONE -"
40 INPUT "": LET riga=9+2*m: LET co
l=v: PRINT AT riga,col;"

"
42 LET c$=STR$ m: GO SUB 9989
45 LET col=u: LET c$=")": GO SUB 99
89
50 LET t=u: GO SUB 900: INPUT a:
LET t=v: GO SUB 900
52 LET t=u: GO SUB 910: IF m<>5
THEN INPUT b
53 LET t=v: GO SUB 910
55 IF m=u THEN INPUT "Numero Sotto
intervalli ";n
60 GO SUB (100 AND m=1)+(200 AND m=
2)+(300 AND m=3)+(400 AND m=4)+(
500 AND m=5)
65 PRINT AT 10+2*m,5; INVERSE u;"RI
SULTATO"; INVERSE v;"=";TAB 18;
STR$ nt: BEEP .1,20: BEEP .1,25:
BEEP .1,20
70 PRINT #u;AT v,v;" PREMI : s PER
UN ALTRO METODO n PER
FINE PROGRAMMA ": BEEP 1,25:
BEEP .5,20
75 PAUSE v: LET t$=CHR$ PEEK 23560:
IF t$="" OR t$<>"n" AND t$
<>"s" THEN GO TO 75
80 IF t$="s" THEN INPUT "":
GO TO 30
90 STOP
100 LET nt=v: LET st=v: LET h=(b-a)/
(2*n): LET n1=2*n-u: LET n2=n1-u

110 FOR i=u TO n1 STEP 2: LET x=a+i*
h: LET st=st+FN f(x): NEXT i
120 LET nt=4*st: LET st=v
130 FOR i=2 TO n2 STEP 2: LET x=a+i*
h: LET st=st+FN f(x): NEXT i
140 LET nt=nt+2*st+FN f(a)+FN f(b):
LET nt=nt*h/3
150 RETURN
200 LET nt=v: LET h=(b-a)/10
210 FOR i=v TO 5: LET j=10-i: LET x=
a+i*h: LET st=FN f(x): LET x=a+j
*h: LET st=st+FN f(x): LET nt=nt
+b(i+1)*st: NEXT i

```

```

240 LET nt=nt*h*5/299376
250 RETURN
300 LET nt=v: LET h=.5*(b-a): LET si
gn=1
310 FOR l=u TO 2
315 FOR i=u TO 6: LET x=sign*g(i)*h:
LET x=x+h+a: LET nt=nt+h(i)*
FN f(x): NEXT i
325 LET sign=-u: NEXT l
340 LET nt=nt*h: RETURN
400 LET h=.5*(b-a): LET nt=FN f(h+a)

440 LET sign=u: FOR l=u TO 2
450 FOR i=u TO 4: LET x=sign*c(i)*h:
LET x=x+h+a: LET nt=nt+FN f(x):
NEXT i
460 LET sign=-u: NEXT l
470 LET nt=2*h*nt/9: RETURN
500 LET nt=v: LET sign=u
510 FOR l=u TO 2
520 FOR i=u TO 6: LET q=sign*g(i)+l:
LET x=a-l+2/q: LET x=FN f(x)/(q
*q): LET nt=nt+h(i)*x: NEXT i
530 LET sign=-u: NEXT l
550 LET nt=nt*2: RETURN
700 BEEP .5,20: PRINT OVER u;AT 5,1
8; FLASH t;"Quale scegli ";
FLASH v;"?";AT 7,18;" [1,2,3,4,
5]": RETURN
705 BEEP .1,25: PRINT AT 2+m,13;"<--
-";AT 2,10; INVERSE u;"Hai scelt
o"; INVERSE v: PAUSE 10:
RETURN
710 BEEP .1,20: PRINT OVER u;AT 2+m
,21;"<="; FLASH t;"GIUSTO";
FLASH v;"?": RETURN
715 BEEP .1,25: BEEP .1,20: PRINT #1
;AT v,v;"PREMI : c PER CORREGGE
RE Enter PER CONT
INUARE "
717 PAUSE v: LET t$=CHR$ PEEK 23560:
IF t$="" OR t$<>"c" AND t$
<>CHR$ 13 THEN GO TO 717
718 IF t$="c" THEN PRINT AT 2+m,13;
" ": LET t=v: GO SUB 710:
BEEP .5,30: GO TO 30
719 RETURN
900 BEEP .2,20: PRINT FLASH t;AT 10
,5;"Inf.": PRINT AT 9+2*m,3;(("
AND t=u)+(STR$ a AND t=v)):
RETURN
910 BEEP .2,25: PRINT AT 10,20;
FLASH t;"Sup. ";AT 9+2*m,18;(("I
nfinite" AND m=5)+(STR$ b AND m
<>5)) AND t=v: RETURN
1000 DIM b(6): FOR i=u TO 6: READ b(i)
: NEXT i
1010 DATA 16067,106300,-48525,272400,
-260550,213684
1020 DIM c(4): FOR i=u TO 4: READ c(i)
: NEXT i
1030 DATA .1679061842,.5287617831,.60
10186554,.9115893077
1040 DIM g(6): DIM h(6)
1050 FOR i=u TO 6: READ g(i): READ h(
i): NEXT i

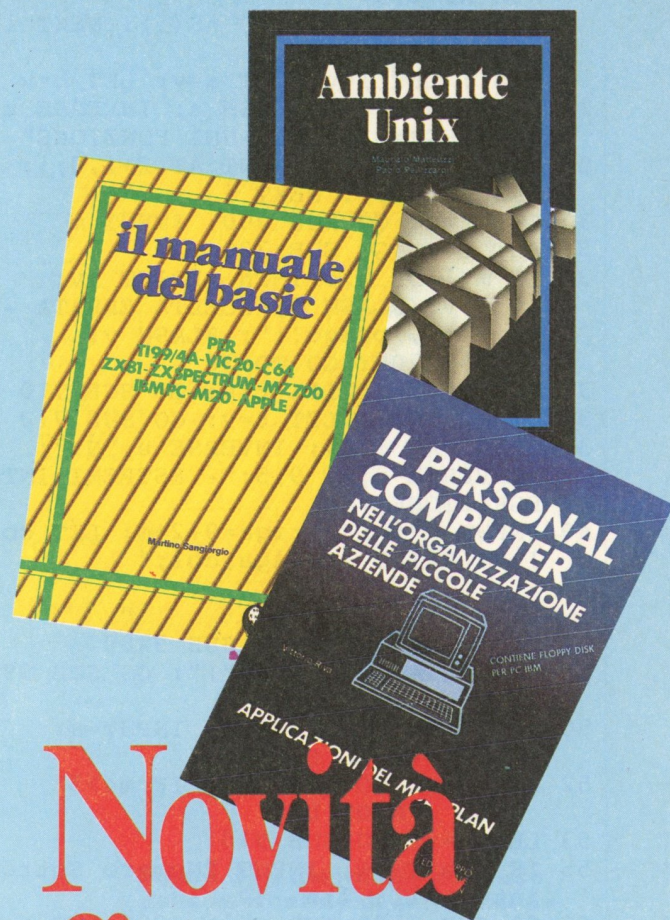
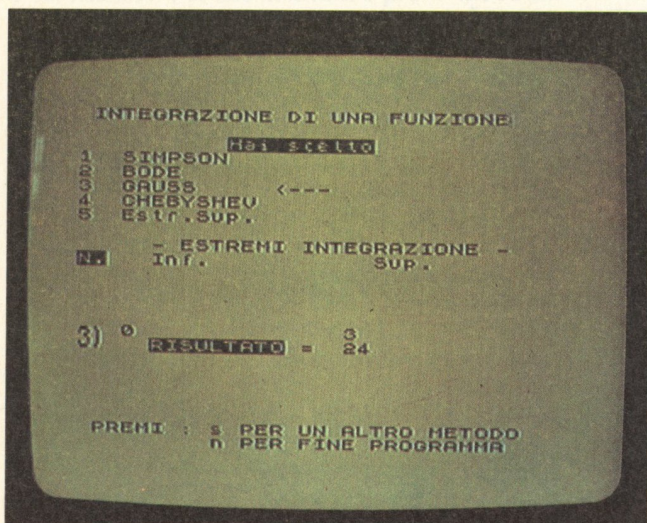
```

LOAD

Integrazione numerica

```
1060 DATA .1252334085,.2491470458,.36
      78314989,.2334925365,.5873179542
      ,.2031674267,.7699026741,.160078
      3285,.9041172563,.1069393259,.98
      15606342,.0471753363
1070 DATA "SIMPSON ","BODE      ","GA
      USS      ","CHEBYSHEV","Estr.Sup."

1100 RETURN
9989 LET col=col-u: FOR c=u TO LEN c$
: GO SUB 9991: PRINT AT riga+u,c
ol+c;"Q";AT riga,col+c;"u":
NEXT c: RETURN
9991 LET z=v: FOR d=v TO 6 STEP 2
9992 POKE USR "u"+d,PEEK (15616+(8*(
CODE c$(c)-32))+z)
9993 POKE USR "u"+d+u,PEEK (15616+(8*(
CODE c$(c)-32))+z)
9994 LET z=z+u: NEXT d
9995 FOR d=u TO 7 STEP 2
9996 POKE USR "q"+d-u,PEEK (15616+8*(
CODE c$(c)-32)+z)
9997 POKE USR "q"+d,PEEK (15616+8*(
CODE c$(c)-32)+z)
9998 LET z=z+u: NEXT d
9999 RETURN
```



Novità firmate Jackson.

IL PERSONAL COMPUTER
NELL'ORGANIZZAZIONE DELLE PICCOLE
AZIENDE: APPLICAZIONI DEL MULTIPLAN
COD. 578P L. 29.000

IL MANUALE DEL BASIC
COD. 534A L. 45.000

AMBIENTE UNIX
COD. 543P L. 19.000



GRUPPO
EDITORIALE
JACKSON

La biblioteca che fa testo.

Latino

di Ferruccio Mandorli

Un freddo, razionale, logico calcolatore può occuparsi anche di lettere classiche!

Chi ha mai detto che i calcolatori non sono utili anche per gli studi umanistici?

Questo programma è un altro esempio della enorme versatilità dei computer in generale e del nostro piccolo grande Spectrum in particolare.

Dopo un mare di programmi in cui siamo stati sommersi da "studi di funzioni" presentati un po' in tutte le salse, ecco ora qualche cosa di nuovo per tutti gli studenti liceali.

Personalmente ho finito il liceo già da alcuni anni, ma credo (anzi tempo) che i professori di latino abbiano ancora la crudele consuetudine di sottoporre i loro malcapitati studenti a una terrificante prova ricorrente: il compito di verbi! Ebbene, "Latino" è un programma che potrà aiutarvi a rendere più divertente la vostra preparazione, conservandone tutta l'efficacia.

Sono messi a vostra disposizione 40 verbi, divisi in 10 per ogni declinazione latina, ed eventualmente sostituibili con semplici modifiche del programma, che lo Spectrum declinerà per voi in tutti i tempi e tutte le persone italiane e latine. Il programma inizia dandovi la possibilità di vedere quali sono i verbi su cui è possibile lavorare, passando poi al primo di 3 menu di scelte:

1° menu

In questa fase avete la possibilità di scegliere tra un verbo in latino oppure in italiano; potete anche rivedere i verbi disponibili oppure terminare l'esecuzione.

Dopo la scelta "italiano" o "latino" il programma vi chiederà l'input del verbo su cui volete lavorare, ovviamente nella lingua già scelta.

Se il verbo richiesto non è in elenco, ciò verrà segnalato dallo Spectrum e avrete la possibilità di rivedere i verbi a disposizione.

In caso contrario vedrete il paradigma del verbo latino, la sua traduzione italiana e la specificazione della declinazione



sia latina che italiana. Si passerà quindi al secondo menu.

2° menu

Avete ora tre possibilità: tornare al menu 1 (per cambiare verbo o per terminare il lavoro), oppure entrare in fase di lettura (dove lo Spectrum stamperà in entrambe le lingue), infine entrare in fase di esercitazione, in cui il computer stamperà nella lingua scelta in partenza e voi dovrete scrivere la traduzione. Immediatamente dopo la scelta "L", o "E", si passerà al terzo ed ultimo menu.

3° menu

Ancora tre scelte a disposizione: una per tornare al menu precedente, una per indicare direttamente il tempo su cui si desidera lavorare (vedi tabella) ed infine un'ultima scelta per la generazione

casuale dei tempi.

Esercitazione

Se siete in fase di esercitazione, lo Spectrum stamperà il verbo declinato nella lingua scelta all'inizio, quindi voi dovrete scrivere la traduzione.

Nello scrivere dovrete seguire le seguenti regole per non incappare in un errore, che abbasserebbe il voto finale.

Non devono essere lasciati spazi bianchi prima della traduzione, non si deve lasciare più di uno spazio bianco tra una parola e la seguente (esempio: "io lodo"), mentre è permesso scrivere tutto attaccato (esempio: "tuavevilodato").

Infine l'infinito futuro va tradotto solo al singolare (esempio: "amaturum, am, um, esse"), ricordandosi le virgole, così come per il participio futuro (esempio: amaturus, a, um).

Noterete poi che l'input è preceduto dalla frase "Verbo 0 per finire": questo perché nel caso in cui le persone da tradurre siano meno di 6 (esempio: gerundio=4, supino=1) è necessario dare zero come input per poter passare alla successiva fase di controllo della traduzione. La fase di controllo è divisa in due parti: la prima, operante subito dopo il comando "C", riscrive la traduzione esatta; la seconda, operante dopo il comando "ENTER", segnala gli errori commessi con un quadrato lampeggiante in corrispondenza della riga in cui è presente l'errore e riscrive la vostra versione errata, in colore rosso, nella riga sottostante alla riga della versione italiana.

Infine vedrete apparire nella parte alta dello schermo, scritto in colore giallo, il vostro voto.

Se avete delle difficoltà di lettura, o se ritenete che i colori usati per la stampa siano poco efficaci, potete modificarli come meglio preferite, andando a modificare poche righe di programma.

Righe degli INK

- 130** INK della stampa dei verbi disponibili
- 1130** INK della stampa dei verbi latini
- 1270** INK della stampa dei verbi italiani
- 1500** INK della stampa dei verbi errati

Funzionamento del programma

Il programma lavora su matrici predefinite, ed è quindi necessario farlo partire con l'auto-start, o con un GOTO 1. Non date RUN, o azzererete le matrici, costringendovi a dover ricaricare il programma!

Le matrici principali sono quelle delle desinenze, degli ausiliari e dei tempi; esse sono strutturate nella seguente maniera: le desinenze (77 per ogni declinazione) sono messe una per riga, in due matrici, una per le desinenze latine e una per quelle italiane.

Nella matrice dei tempi sono invece messi i riferimenti alle righe relative alle desinenze desiderate; facciamo un esempio:

1) indicando come tempo "indpr" il programma andrà a cercare nella matrice dei tempi la riga "indpr 0106", la quale indica che le desinenze dell'indicativo presente si trovano nella matrice delle desinenze a partire dalla riga $1+77*(d-1)$ fino alla riga $6+77*(d-1)$, dove d indica la declinazione del verbo in osservazione.

In coefficiente $77*(d-1)$ serve a "scivolare" sotto alle declinazioni precedenti, come occorre fare per esempio con un verbo della 3^a declinazione.

Nel caso in cui il tempo richiesto sia un tempo composto, al posto della desinenza si troverà un nuovo riferimento per la matrice degli ausiliari; vediamo un nuovo esempio;

Descrizione delle variabili:

IS(40,25)	matrice contenente l'infinito, il participio e l'indicazione del verbo ausiliare dei verbi disponibili
LS(40,30)	matrice contenente il paradigma e la declinazione dei verbi latini disponibili
TS(19,9)	matrice dei tempi
FS(50,8)	matrice degli ausiliari
S(308,7)	matrice delle desinenze latine
NS(231,8)	matrice delle desinenze italiane
FS(10,31)	matrice usata per i controlli (non è predefinita)
RS	stringa contenente la radice italiana del verbo
US	stringa contenente il participio italiano del verbo
SS	stringa contenente la radice del presente del verbo latino
QS	stringa contenente la radice del passato del verbo latino
CS,BS,K\$,P\$,VS,AS,JS	stringhe di uso vario
VI	variabile della declinazione italiana
VL	variabile della declinazione latina

Descrizione del programma

Linee	Commento
15	POKE per abilitare il nuovo set di caratteri grafici e per aumentare il click della tastiera
20	dimensionamento B\$ e C\$
79-150	routine di stampa dei verbi disponibili
160	menu 1
180	stop con ripristino del set di caratteri originali
240	input verbo italiano
280	dimensionamento R\$
290	riconoscimento della radice del verbo italiano
300-320	calcolo della declinazione italiana
340-370	controllo della disponibilità del verbo richiesto
400-420	stampa paradigma verbo latino, infinito verbo italiano, declinazione latina, declinazione italiana
470	dimensionamento e assegnazione U\$
500	input verbo latino
530-560	controllo disponibilità verbo richiesto
620-640	calcolo declinazione
650	dimensionamento ed assegnazione U\$
660-680	stampa paradigma, infinito, participio e declinazione dei verbi italiano e latino
690	dimensionamento R\$
700	assegnazione R\$
760	assegnazione S\$
790	menu 2
840	menu 3
890	input tempo
950-970	ricerca del tempo nella matrice dei tempi
990-1090	assegnazione C\$, J\$, P\$
1110-1150	stampa latino
1200-1290	stampa italiano
1380-1460	controllo della traduzione
1470	normalizzazione voto e sua stampa
1480-1510	stampa errori
1520	dimensionamento F\$
1530-1570	input traduzioni
1620-1670	ricerca ausiliare
1680-1720	segnalazione errori e calcolo voto
1760-1900	ricerca radice del passato del verbo latino

2) richiedendo il tempo "indpp" il programma troverà la riga "indpp 2530", che rimanderà alle righe dalla 25 alla 30 (sempre moltiplicate per il coefficiente $77*(d-1)$, dove si troveranno i valori 01,02,03, ecc. i quali indicano che nelle righe 1,2,3, ecc. della matrice degli ausiliari trovano: avevo, avevi, aveva, ecc. oppure ero, eri, era, ecc..

Precedentemente il programma si era preoccupato di riconoscere le radici dei verbi, sia latini che italiani, a cui ora aggiungerà le desinenze trovate nel modo suddetto, per poi eseguire la stampa completa.

Per quanto riguarda casi particolari, come per esempio l'infinito futuro, il comportamento del programma è legger-

mente diverso, ma potrete farvene un'idea dando un'occhiata alle specifiche di programma e al listato, che è stato ordinato con il programma Spectrum renumber, apparso sul numero 3 di questa stessa rivista, e che appare quindi abbastanza chiaro.

Modifica caratteri

Avrete certamente notato che il programma funziona con un nuovo set di caratteri grafici. Cercherò ora di spiegare come ciò è stato possibile.

Innanzitutto occorre sapere che lo Spectrum ha in dotazione un set di 127 caratteri, collocati nella memoria ROM a partire dalla locazione 15360. I primi 31 caratteri sono però di controllo, e quindi inutilizzabili dall'utente; ne segue che il set è praticamente formato da 96 caratteri, a partire dallo spazio bianco fino al carattere di copyright, memorizzati nella ROM a partire dalla locazione 15616, fino alla locazione 16386.

Lo Spectrum è poi dotato di una variabile di sistema chiamata CHARS (vedere anche il manuale) che contiene il valore della locazione di inizio del set di caratteri.

È quindi possibile creare un nuovo set di caratteri e porlo in memoria RAM, per poi modificare il valore di CHARS, adeguandolo alla posizione di questo nuovo set grafico. Nel nostro caso i caratteri modificati sono solamente i numeri e le maiuscole. Per fare questo ho operato nel modo seguente:

ho trasferito tutto il set originale (lungo 768 byte) in una nuova zona della RAM libera (quella a partire dall'indirizzo 53248) ed ho poi modificato i numeri e le minuscole, cambiando i valori delle locazioni da 53768 a 53975 per le minuscole, e da 53376 a 53455 per i numeri.

Ho quindi creato un nuovo set grafico a partire dalla locazione 53248, e per abilitarlo ho modificato CHARS con due POKE del tipo seguente:

POKE 23606,a POKE 23607,b

dove:

$a = (53248 - 256) - b * 256 = 0$

$b = \text{INT}((53248 - 256) / 256) = 207$

(provate a battere POKE 23606,0: POKE 23607,207 prima di listare il programma).

Per tornare al set originale è invece necessario riportare CHARS al suo primitivo valore, effettuando

POKE 23606,0:POKE 23607,60

Per salvare il nuovo set grafico bisogna digitare

SAVE "CODE 53248,768

Inserimento nuovi verbi

Potete sostituire dei verbi, oppure inserirne dei nuovi, modificando o rifacendo totalmente le matrici dei verbi, tenendo conto delle seguenti limitazioni:

1) le matrici devono chiamarsi I\$ per i

Tabella dei tempi

INDPR	indicativo presente
INDIM	indicativo imperfetto
INDFS	indicativo futuro semplice
INDPE	indicativo perfetto
INDPP	indicativo piucheperfecto
INDFA	indicativo futuro anteriore
CONPR	congiuntivo presente
ONIM	congiuntivo imperfetto
CONPE	congiuntivo perfetto
CONPP	congiuntivo piucheperfecto
INFPR	infinito presente
INFPE	infinito perfetto
INFFT	infinito futuro
IMPPR	imperativo presente
IMPFS	imperativo futuro
PARPR	participio presente
PARFT	participio futuro
GER	gerundio
SUP	supino

È bene che ognuno di voi ricordi che alcuni tempi latini hanno più di una traduzione italiana; la scelta di un tempo piuttosto che un altro (esempio: il congiuntivo piuttosto che il condizionale) è

del tutto arbitraria, come quella di identificare i tempi con la loro espressione latina (esempio: perfetto piuttosto che passato remoto).

Valore decimale del nuovo set di caratteri

Numeri

"0"	0 60 102 110 118 102 60 0
"1"	0 28 60 108 12 12 12 0
"2"	0 126 70 6 62 64 126 0
"3"	0 124 6 60 6 6 124 0
"4"	0 6 14 22 38 127 6 0
"5"	0 126 96 124 6 6 124 0
"6"	0 56 96 124 102 102 60 0
"7"	0 126 126 6 12 24 48 0
"8"	0 60 102 60 102 102 60 0
"9"	0 60 102 102 62 6 60 0

Lettere

"a"	0 126 98 98 126 98 98 0
"b"	0 126 98 126 98 98 126 0
"c"	0 126 96 96 96 96 126 0
"d"	0 124 98 98 98 98 124 0
"e"	0 126 96 124 96 96 126 0
"f"	0 126 96 124 96 96 96 0
"g"	0 126 96 96 110 98 126 0
"h"	0 98 98 126 98 98 98 0
"i"	0 24 24 24 24 24 24 0
"j"	0 6 6 6 38 38 62 0
"k"	0 104 112 96 112 104 100 0
"l"	0 96 96 96 96 96 126 0
"m"	0 98 118 106 98 98 98 0
"n"	0 98 98 114 106 102 98 0
"o"	0 126 98 98 98 98 126 0
"p"	0 126 98 126 96 96 96 0
"q"	0 126 98 98 106 106 126 4
"r"	0 126 98 126 104 100 98 0
"s"	0 126 96 126 2 2 126 0
"t"	0 126 24 24 24 24 24 0
"u"	0 98 98 98 98 98 126 0
"v"	0 98 98 100 100 104 56 0
"w"	0 66 66 90 90 126 36 0
"x"	0 98 52 24 28 38 66 0
"y"	0 66 38 28 24 48 96 0
"z"	0 126 126 8 16 32 126 0

verbi italiani e L\$ per i verbi latini, e devono avere le dimensioni I\$(40,25), L\$(40,30);

2) alla stessa riga delle due matrici deve corrispondere lo stesso verbo (esempio: I\$(1) = "amare, amato 0", L\$(1) = "lamo,as,avi,atum,are");

3) nella matrice I\$ vanno inseriti per ogni riga, a partire dalla prima colonna, l'infinito e il participio del verbo, intervallati da una virgola.

Alla colonna 25 di ogni verbo si deve

inserire "0", se il suo ausiliare è "avere", e "1", se il suo ausiliare è "essere";
4) nella matrice L\$ vanno inseriti in ogni riga i paradigmi dei verbi scritti tutti di seguito ed intervallati da virgole, tenendo presente che se il perfetto ha una desinenza irregolare, si deve scrivere anche la radice.

Inoltre si deve inserire nella prima colonna il numero corrispondente alla declinazione latina del verbo (esempio: "lamo,as,avi,atum,are", "3cedo,is,ces-

si,cessum,ere").

Ricordarsi che per rispettare le dimensioni delle matrici la lunghezza dei paradigmi non deve superare i 29 caratteri, virgole comprese.

È inutile ricordare che per il buon funzionamento del programma i verbi inseriti devono essere regolari, sia in latino che in italiano.

A questo punto credo proprio di avervi detto tutto, e quindi non mi resta che augurarvi ...buon studio!!!

```

10 PAPER 1: INK 7: BORDER 1:
   BRIGHT 1: CLS
15 POKE 23606,0: POKE 23607,207:
   POKE 23608,50: POKE 23609,50
20 DIM b$(4): DIM c$(3)
30 PRINT #0; INK 5; AT 0,0; "vuoi vedere i verbi"; AT 1,0; "a disposizione ? (s/n)"
40 IF INKEY$="n" THEN CLS :
   GO TO 160
50 IF INKEY$="s" THEN LET l=0:
   LET a=1: LET b=20: GO TO 70
60 GO TO 40
70 CLS : PRINT AT 0,5; FLASH 1;
   INK 5; "verbi a disposizione "
   ': FOR i=a TO b
80 FOR k=1 TO 29
90 FOR L$(i,k)="," THEN LET l=l+1:
   IF l=2 THEN LET l=0: GO TO 110

100 NEXT k
110 FOR o=1 TO 25: IF I$(i,o)=","
   THEN GO TO 130
120 NEXT o
130 PRINT INK 4; I$(i, TO o-1),
   INK 7; "=" ; INK 3; L$(i,2 TO k);
   "..": NEXT i
140 PRINT #0; AT 1,0; INK 5; "Premi enter": PAUSE 0: CLS
150 IF a=1 THEN LET a=21: LET b=40:
   GO TO 70
160 PRINT AT 0,11; FLASH 1; INK 6; "
   menu' 1 "
170 PRINT "" scegli:"; INK 4; "
   i=italiano"" INK 3; "
   l=latino"" INK 5; "
   v=visione verbi"" INK 6; "
   9=fine lavoro"

180 IF INKEY$="9" THEN POKE 23606,0
   : POKE 23607,60: STOP
190 IF INKEY$="i" THEN LET m=1:
   GO TO 230
200 IF INKEY$="l" THEN LET m=2:
   GO TO 490
210 IF INKEY$="v" THEN LET l=0:
   LET a=1: LET b=20: GO TO 70
220 GO TO 180
230 CLS
240 PRINT AT 1,1; INK 4; "input verbo
   all'infinito"
250 INPUT v$

```

```

260 IF LEN v$<4 THEN GO TO 250
270 LET vi=0: LET vl=0
280 DIM r$(LEN v$-3)
290 LET r$=v$(1 TO LEN v$-3)
300 IF v$(LEN v$-2)="a" THEN LET vi
   =1
310 IF v$(LEN v$-2)="e" THEN LET vi
   =2
320 IF v$(LEN v$-2)="i" THEN LET vi
   =3
330 IF vi=0 THEN GO TO 250
340 FOR i=1 TO 40
350 IF v$=I$(i, TO LEN v$) THEN
   GO TO 370
360 NEXT i
370 IF i>40 THEN PRINT AT 19,0; "il
   verbo "; FLASH 1; INK 4; v$;
   FLASH 0; INK 7; "non e' a dispos
   izione": GO TO 30
380 CLS
390 LET vl=VAL L$(i,1)
400 PRINT INK 4; v$; '' INK 3; L$(i,2
   TO 30)
410 PRINT ''
420 PRINT INK 3; vl; '' declinazione
   latina"; '' ; INK 4; vi; '' declinaz
   ione italiana"
430 FOR j=1 TO 25
440 IF I$(i,j)="," THEN LET a=j+1
450 IF I$(i,j)=" " THEN LET b=j-1:
   GO TO 470
460 NEXT j
470 DIM u$(b-a+1): LET u$=I$(i,a
   TO b)
480 GO TO 710
490 CLS
500 PRINT AT 1,1; INK 3; "input l' pe
   rsona singolare"" presente del
   verbo voluto"
510 INPUT v$
520 IF v$="" THEN GO TO 510
530 FOR i=1 TO 40
540 IF v$=L$(i,2 TO LEN v$+1) THEN
   GO TO 570
550 NEXT i
560 IF i>40 THEN PRINT AT 19,0; "il
   verbo "; FLASH 1; INK 3; v$;
   FLASH 0; INK 7; "non e' a dispo
   sizione": GO TO 30
570 LET vl=VAL L$(i,1)
580 FOR j=1 TO 25
590 IF I$(i,j)="," THEN LET a=j

```

```

600 IF I$(i,j)=" " THEN LET b=j-1:
    GO TO 620
610 NEXT j
620 IF I$(i,a-3)="a" THEN LET vi=1
630 IF I$(i,a-3)="e" THEN LET vi=2
640 IF I$(i,a-3)="i" THEN LET vi=3
650 DIM u$(b-a): LET u$=I$(i,a+1
    TO b)
660 CLS : PRINT INK 3;L$(i,2 TO 25)
    ;'' INK 4;I$(i,1 TO j-1)
670 PRINT ''
680 PRINT INK 3;v1;'' declinazione
    latina''; INK 4;vi;'' declinazi
    one italiana"
690 DIM r$(a-4)
700 LET r$=I$(i,1 TO a-4)
710 LET z=i
720 FOR f=1 TO 30
730 IF L$(i,f)="," THEN GO TO 750
740 NEXT f
750 LET l=f-2-(ABS SIN (PI/2*(VAL L
    $(i,l)-1)))
760 LET s$=L$(i,2 TO 1)
770 GO SUB 1750
780 PRINT #0; INK 5;AT 0,0;"premi en
    ter": PAUSE 0
790 CLS : PRINT AT 0,10; INK 6;
    FLASH 1;" menu' 2": PRINT '';"
    scegli:"; INK 6;" 1 per lettu
    ra e per eserc
    itazione 0 per menu'
    1"
800 IF INKEY$="0" THEN CLS :
    GO TO 160
810 IF INKEY$="1" THEN LET x=1:
    GO TO 840
820 IF INKEY$="e" THEN LET x=0:
    LET c=0: GO TO 840
830 GO TO 800
840 CLS : PRINT AT 0,10; FLASH 1;
    INK 6;" menu' 3": PRINT '';"
    scegli:"; INK 5;" t=tempo indic
    ato v=tempi casua
    li 0=menu' 2"
850 IF INKEY$="t" THEN GO TO 890
860 IF INKEY$="v" THEN GO TO 1730
870 IF INKEY$="0" THEN GO TO 790
880 GO TO 850
890 INPUT "tempo ";a$
900 DIM c$(3)
910 LET vot=10
920 CLS
930 PRINT AT 0,15; INK 5;"tempo: ";a
    $
940 PRINT AT 0,0; INK 3;"latino":
    PRINT AT 1,4; INK 4;"italiano":
    PLOT 0,158: DRAW 255,0
950 FOR i=1 TO 19
960 IF a$=t$(i, TO LEN a$) THEN
    GO TO 990
970 NEXT i
980 IF i>=19 THEN GO TO 890
990 IF t$(i,1)="s" THEN LET c$="a":
    GO TO 1030
1000 IF t$(i, TO 5)="infft" THEN
    DIM c$(9): LET c$="stare per":
    GO TO 1030
1010 IF t$(i,1)="c" OR t$(i,1)="p"
    THEN LET c$="che": GO TO 1030
1020 LET c$=" "
1030 IF m=1 THEN GO TO 1180
1040 LET d$=t$(i,4 TO 5)
1050 IF t$(i,1)="s" THEN DIM p$(
    LEN w$): LET p$=w$: GO TO 1100
1060 IF t$(i, TO 5)="parft" THEN
    DIM p$(LEN w$): LET p$=w$:
    LET j$="urus,a,um": LET t=1:
    GO TO 1100
1070 IF t$(i, TO 5)="infft" THEN
    DIM p$(LEN w$): LET p$=w$:
    LET j$="urum,am,um esse": LET t
    =1: GO TO 1100
1080 IF d$="pe" OR d$="pp" OR d$="fa"
    THEN DIM p$(LEN q$): LET p$=q$
    : GO TO 1100
1090 IF d$="pr" OR d$="im" OR d$="fs"
    THEN DIM p$(LEN s$): LET p$=s$
1100 LET k=1
1110 FOR j=VAL t$(i,6 TO 7)+(v1-1)*77
    TO VAL t$(i,8 TO 9)+(v1-1)*77
1120 IF t=1 THEN PRINT AT k*3,0;
    INK 3;p$;j$: LET t=0: GO TO 114
    0
1130 PRINT AT k*3,0; INK 3;p$;z$(j)
1140 LET k=k+1
1150 NEXT j
1160 IF x=0 AND c=0 THEN GO SUB 1520
1170 IF m=1 THEN GO TO 1330
1180 LET j=VAL t$(i,6 TO 7)+(vi-1)*77
1190 LET w=0: LET k=1
1200 FOR j=VAL t$(i,6 TO 7)+(vi-1)*77
    TO VAL t$(i,8 TO 9)+(vi-1)*77
1210 LET w=w+1
1220 LET b$=m$(k)
1230 IF j-(77*(vi-1))>60 THEN LET b$
    =" "
1240 DIM p$(8): LET p$=n$(j): LET k$=
    r$
1250 IF n$(j,1)=" " THEN GO SUB 1620
1260 IF t$(i,1)="g" THEN LET c$=g$(w
    )
1270 PRINT AT k*3+1,0; INK 4;c$;" ";b
    $;" ";k$;p$
1280 LET k=k+1
1290 NEXT j
1300 LET k$=r$
1310 IF x=0 AND c=0 THEN GO SUB 1520
1320 IF m=1 THEN GO TO 1040
1330 PRINT #0;AT 0,0; INK 5;"premi en
    ter": PAUSE 0
1340 IF x<>0 THEN GO TO 840

```

LOAD

Latino

```

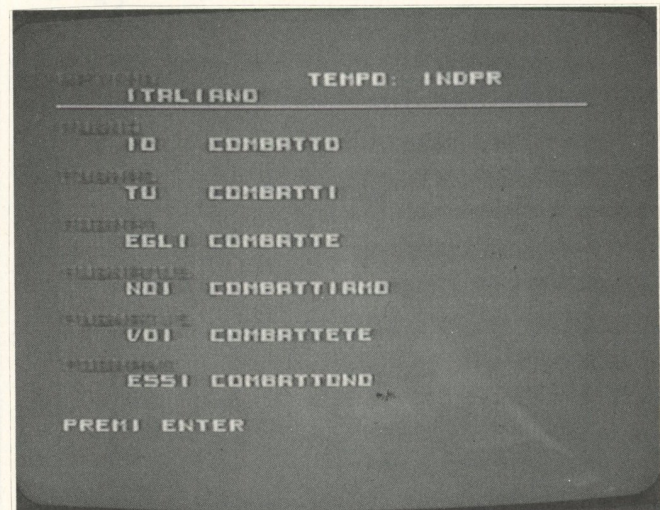
1350 DIM e(10)
1360 PRINT #0;AT 0,0;"
      "
1370 LET d=1
1380 FOR k=1 TO q-1
1390 LET s=1
1400 FOR y=0 TO 30
1410 IF SCREEN$(k*3+m-1,y)=" "
      THEN NEXT y
1420 IF f$(k,s)=" " THEN LET s=s+1
1430 IF SCREEN$(k*3+m-1,y)<>f$(k,s
      ) THEN GO TO 1680
1440 LET s=s+1
1450 NEXT y
1460 NEXT k
1470 PRINT AT 1,15; INK 6;"voto: ";
      INT (vot*100)/100
1480 FOR d=1 TO 10
1490 IF e(d)=0 THEN PRINT #0;AT 0,0;
      INK 5;"premi enter": PAUSE 0:
      LET c=0: GO TO 840
1500 PRINT AT (e(d))*3+2,0; INK 2;f$(
      e(d))
1510 NEXT d
1520 LET c=1: DIM f$(10,31)
1530 FOR q=1 TO 6
1540 INPUT INK 5;"verbo 0=fine ";f$(
      q)
1550 IF f$(q,1)="0" THEN GO TO 1580
1560 PRINT AT q*3+m-1,0;f$(q)
1570 NEXT q
1580 PRINT #0;AT 0,0; INK 5;"c per co
      ntrollo"
1590 IF INKEY$<>"c" THEN GO TO 159
      0
1600 PRINT #0;AT 0,0;"
      "
1610 RETURN
1620 REM ausil
1630 DIM p$(LEN u$)
1640 LET p$=u$
1650 IF I$(z,25)="1" AND k>2 THEN
      LET p$(LEN u$)="i"
1660 LET k$=e$((VAL n$(j,2 TO 3))+
      VAL I$(z,25)*25), TO 8)+" "
1670 RETURN
1680 PRINT AT k*3+m-1,31; FLASH 1;"
      {SG8}"
1690 LET vot=vot-10/(q-1): IF vot<0
      THEN LET vot=0
1700 LET e(d)=k
1710 LET d=d+1
1720 GO TO 1460

```

```

1730 LET a$=t$(INT (RND*19)+1, TO 5)
1740 GO TO 900
1750 LET w=0: LET c=0
1760 FOR h=1 TO 25
1770 IF L$(i,h)="," THEN LET c=c+1
1780 IF c=2 THEN LET w=h: LET c=3
1790 IF c=4 THEN GO TO 1810
1800 NEXT h
1810 LET h=h-w-1
1820 IF h>3 THEN LET q$=L$(i,w+1
      TO w+h-1): GO TO 1840
1830 LET q$=s$+L$(i,w+1 TO w+h-1)
1840 LET w=w+h+2: FOR h=w TO 30
1850 IF L$(i,h)="," THEN GO TO 1870
1860 NEXT h
1870 LET h=h-w
1880 IF h>4 THEN LET w$=L$(i,w TO w+
      h-3): RETURN
1890 LET w$=s$+L$(i,w TO w+h-3)
1900 RETURN

```



Dietro un bambino maltrattato c'è sempre una famiglia in crisi.

È impossibile stabilire con esattezza quanti bambini in Italia subiscono violenze. Tanti, purtroppo. Ma una cosa è certa: la violenza che oggi un bambino subisce e impara, da grande la ritorcerà contro la società.

Bisogna intervenire, e subito: agendo sulle cause. Intervendendo sulla crisi familiare che è sempre alla base dei maltrattamenti a un bambino.

**UN BAMBINO
MALTRATTATO OGGI
SARÀ UN UOMO
VIOLENTO DOMANI.
FACCIAMO QUALCOSA
PRIMA.**

**"Vorrei contribuire anch'io".
Per sostenere il C.A.F. vi invio**

- L. 10.000 L. 25.000 L. 50.000
 L. 100.000 L. 500.000 L.

- Tramite assegno bancario intestato C.A.F.
Via V. E. Orlando 15, 20142 Milano, che vi invio,
unito a questo tagliando.
 Tramite versamento sul C.C.P. n° 22349203 intestato C.A.F.
Via V. E. Orlando 15, 20142 Milano.

Nome e Cognome.....

Indirizzo

CAP..... Città

Provincia

Attendo: ricevuta e tesserino di "Sostenitore" del C.A.F.

JAR

Segnaliamo ai Servizi Sociali del Territorio i casi di maltrattamento.

La prima cosa da fare: segnalare ogni caso di maltrattamento di cui siamo a conoscenza o di cui abbiamo il fondato sospetto.

Telefonate ai Servizi Sociali del Territorio dal lunedì al venerdì,

dalle ore 9 alle ore 12 (troverete i numeri di telefono su Tuttocittà), oppure al Tribunale per i Minorenni (chiedendo del giudice di turno) dal lunedì al sabato dalle ore 9 alle ore 13, telefono 4988001/2/3: oppure al C.A.F., a questi numeri di telefono 02/8265051/2, che rispondono 24 ore su 24, 7 giorni su 7.

Sosteniamo il C.A.F.

Il C.A.F. è il primo Centro di Aiuto al Bambino Maltrattato e alla Famiglia in Crisi, sorto in Italia con lo scopo di intervenire nei casi di maltrattamento.

Oggi, nella sua nuova sede più grande e attrezzata, accoglie provvisoriamente i bambini maltrattati, contemporaneamente svolge un'azione terapeutica sulla famiglia per ripristinare l'equilibrio e reintegrare il bambino nel suo ambiente.

Il C.A.F. è un'istituzione privata che opera in collaborazione con i Servizi Sociali, con i supporti messi gratuitamente a disposizione dal Comune di Milano, e con i contributi finanziari volontari dei cittadini.

Contribuisci anche tu, utilizzando il tagliando. È un gesto necessario, per i bambini maltrattati, per te che vuoi fare qualcosa di concreto contro la violenza.

C.A.F.



**CENTRO DI AIUTO
AL BAMBINO MALTRATTATO
E ALLA FAMIGLIA IN CRISI**

Classifiche dei campionati

di Gianni Marion

Per poter gestire le classifiche dei campionati di calcio basta uno Spectrum 16K!

Questo programma permette di aggiornare le classifiche dei campionati di calcio nazionali, dalla serie A alla C2, fornendo per ogni squadra 23 differenti informazioni su punteggi e dati di possibile interesse, per un totale di 3312 informazioni.

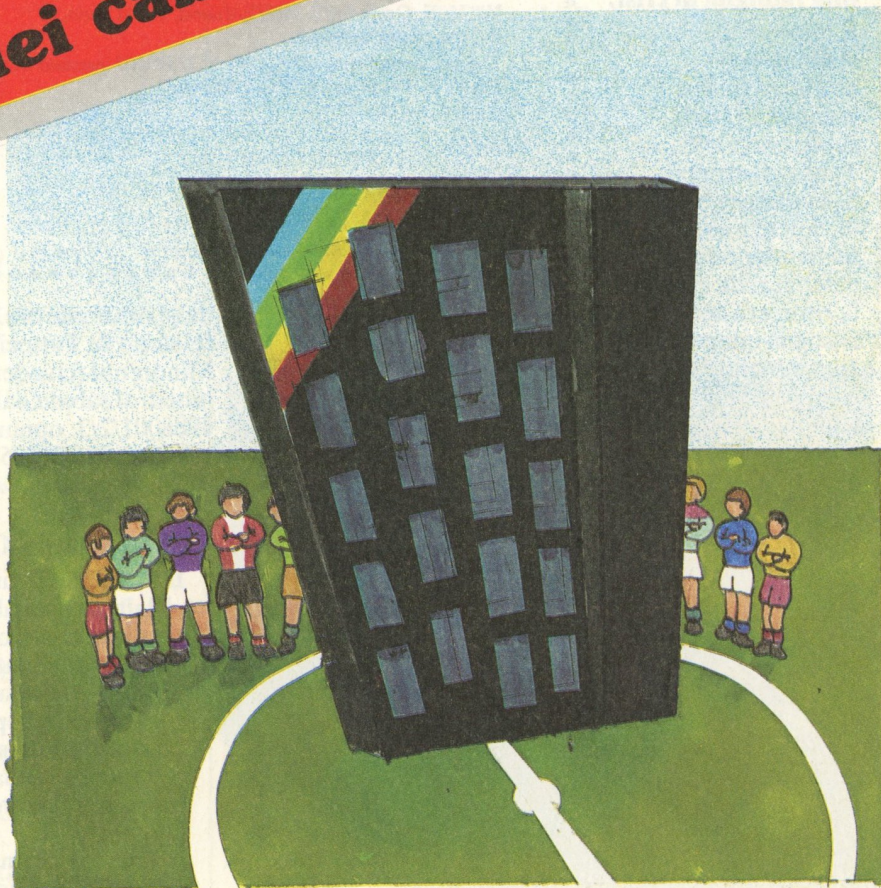
All'inizio del programma sono presentate 6 opzioni, che permettono di aggiornare, correggere o visualizzare le classifiche e di caricare o salvare i dati relativi alle varie giornate e i nomi delle squadre.

Esiste inoltre una procedura di "inizializzazione" che permette di azzerare la classifica della serie prescelta, predisponendosi a ricevere i nominativi delle squadre (operazione che dovrà essere effettuata all'inizio del campionato).

Le opzioni che vengono presentate in ogni stadio del programma sono ottenibili premendo il tasto corrispondente, riportato a fianco dell'istruzione in INVERSE.

Il programma è in un loop continuo e riparte senza difficoltà dalla linea 8995. Per salvarlo basta digitare RUN 9999. "Classifiche dei campionati" utilizza quattro routine in linguaggio macchina di 474 byte (alloggiati parte nel buffer della stampante e parte sopra RAMTOP per esigenze di spazio) che vengono generate mediante il programma "machine code" e poi salvate in coda al programma principale (sulla cassetta, comunque, troverete il programma già funzionante).

In particolare una delle routine ordina le classifiche secondo uno dei 20 para-



Descrizione del programma

Linee

1-55

1-30

35-40

45-55

1000-70

1010-30

1040

1040-70

2000-4020

2000

3000

4000

5000-50

5000

5010-20

5030-50

6000

6000

7000

8000-40

8000-100

Commento

AGGIORNAMENTO

controlla il cursore

seleziona le squadre

riceve il risultato e aggiorna le classifiche

SELEZIONE SERIE

inizializza le variabili caratteristiche

predisporre i parametri per le routine L/M

ritorno subroutine

CARICA E SALVA I DATI

scelta nomi o classifiche

salva

carica

CORREZIONE ERRORI/ANDAMENTO IN CASA E TRASFERTA

scelta squadra

scelta dato da correggere

correzione

INIZIALIZZAZIONE

azzerare la classifica e ricevere i nomi delle squadre della serie prescelta

INTESTAZIONE

VISUALIZZA LE CLASSIFICHE

scelta classifica

metri a scelta (linea 8110), un'altra visualizza le classifiche con i riepiloghi generali (linea 8120), una terza esegue l'aggiornamento dei dati (linea 55) e l'ultima azzerava gli spazi destinati alle classifiche (linea 6000).

I dati sono memorizzati a partire dalla locazione 31040 nel seguente ordine per ogni squadra: vinte, pareggiate, perse, reti fatte e subite in casa, e di seguito, nello stesso ordine, per le partite in trasferta; è poi memorizzato il totale dei punti delle ultime cinque giornate e quelli delle ultime quattro moltiplicati ciascuno per un opportuno fattore. Aggiornando le classifiche, appare nella parte bassa dello schermo un cursore, che può essere posizionato sulla squa-

81102
8120-40
8990

ordina la classifica e intesta la pagina
visualizza la classifica

8995
9000-100
9010-100
9999

START
carica i codici L/M e sposta la RAMTOPO
INIZIALIZZAZIONE VARIABILI PRINCIPALI
MENÙ GENERALE
scelta opzione
SALVA IL PROGRAMMA

dra desiderata con le frecce 6 e 7 (6 lo sposta sulla squadra successiva, 7 su quella precedente).

Una volta rintracciata la squadra che gioca in casa si deve premere ENTER e ripetere l'operazione sulla squadra in trasferta, dopodiché sarà sufficiente inserire il risultato della partita, per il qua-

le non viene visualizzato nessun cursore (non pensate che il programma si sia bloccato!).

Nella routine di correzione quando vengono richiesti i punteggi delle ultime 5 giornate bisogna inserirli consecutivamente, cominciando dal meno recente.

Classifiche dei campionati versione per ZX Spectrum 16K/48K programma principale

```

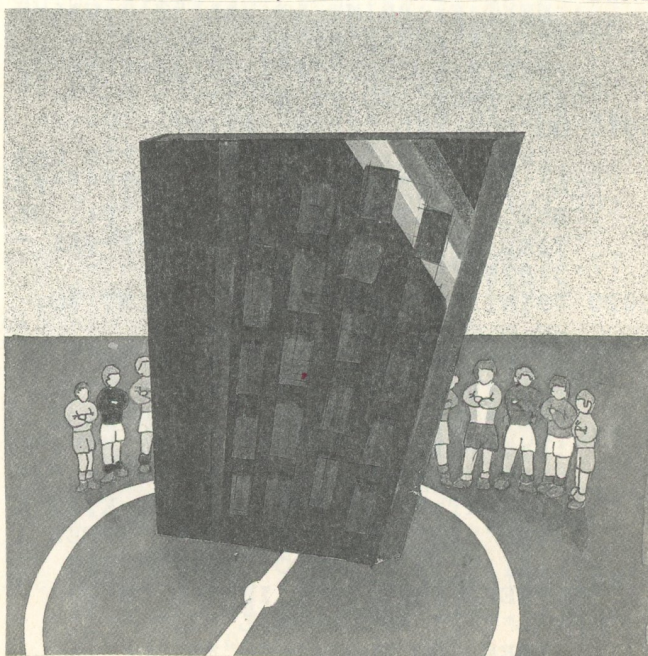
1 GO SUB VAL "7"*W: PRINT "SERIE "
;S$;AT VAL "16",0;" "; INVERSE X
;"6";: PRINT " AVANTI",
  INVERSE X;"7";: PRINT " INDIETR
O"" ENTER per selezionare"" -R
- in caso di errore""P$
3 LET L=0: FOR F=X TO R/VAL "2"
5 LET L=L+X*(L<R): PRINT #0;AT X,0
;A$(K+X);" - ";A$(K+X);" ":
  LET J=0: LET G=VAL "35"
10 IF INKEY$="U" THEN GO TO W
15 LET I=K+L: PRINT #0; FLASH X;
  AT X,J;A$(I): PAUSE 0: IF
  INKEY$=CHR$ VAL "13" THEN
  GO TO G
20 IF INKEY$="6" THEN LET L=L+X-R*
(L=R)
25 IF INKEY$="7" THEN LET L=L+-X+R
*(L=X)
30 GO TO B-X-X
35 POKE VAL "23318",I: LET G=VAL "4
0": LET J=B+X: PRINT #0;AT X,0;A
$(I);" - ";: LET L=X: GO TO B-X-
X
40 POKE VAL "23307",I: PRINT #0;
  AT X,J;A$(I)
45 PAUSE 0: LET J=CODE INKEY$:
  IF J<CODE "0" OR J>CODE "9"
  THEN LET L=0: GO TO VAL "5"
50 PRINT #0;AT X,B+B;CHR$ J;"-";:
  POKE VAL "23331",VAL CHR$ J:
  PAUSE 0: LET J=CODE INKEY$:
  IF J<CODE "0" OR J>CODE "9"
  THEN PRINT #0;AT X,B+B;" ":
  GO TO VAL "45"
55 PRINT #0;CHR$ J: IF J>CODE "/"
  AND J<CODE ":" THEN POKE VAL "
23332",VAL CHR$ J: RANDOMIZE
  USR VAL "23306": LET L=PEEK
  VAL "23318": PRINT AT F+X+X,0;A$

```

```

(L);" - ";A$(PEEK VAL "23307");"
";PEEK VAL "23331";"-";PEEK
VAL "23332": LET L=L-K: NEXT F:
  PRINT #0;AT X,0;C$: PAUSE 0:
  CLS : GO TO W*(INKEY$="U")
60 GO TO VAL "55"
1000 GO SUB VAL "7"*W: IF L$="I"
  THEN PRINT "SQUADRE"
1005: PRINT "" "; INVERSE X;X;:
  PRINT " SERIE A"" ";
  INVERSE X;"2";: PRINT " SERI
E B"" "; INVERSE X;"3";:
  PRINT " SERIE C1 GIRONE A""
"; INVERSE X;"4";: PRINT "
SERIE C1 GIRONE B"" ";
  INVERSE X;"5";: PRINT " SERI
E C2 GIRONE A"" "; INVERSE X;"
6";: PRINT " SERIE C2 GIRONE
B"" "; INVERSE X;"7";: PRINT "
SERIE C2 GIRONE C"" ";
  INVERSE X;"8";: PRINT " SERI

```



```

E C2 GIRONE D''''P$: PAUSE O:
  LET J=CODE INKEY$-VAL "48":
  IF J=VAL "37" THEN GO TO VAL "
  9"*W
1010 LET R=VAL "16": IF J<=X OR J>
  VAL "8" THEN LET K=O: LET S$="A
  ": GO TO VAL "1040"
1020 IF J=X+X THEN LET K=R: LET R=
  VAL "20": LET S$="B"
1030 IF J>X+X THEN LET R=VAL "18":
  LET K=R*(J-X): LET S$=N$(X+(B-X
  )*(J-VAL "3") TO (B-X)*(J-X-X))
1040 POKE VAL "23297",K: POKE VAL "30
  856",R: IF L$="C" THEN GO TO
  VAL "8"*W
1050 IF L$="I" THEN RETURN
1060 IF L$="E" THEN GO TO VAL "5"*W
1070 CLS : GO TO O
2000 GO SUB VAL "7"*W: PRINT "' ' " ;
  INVERSE X;"N";: PRINT "  NOMI
  SQUADRE"' ' " ; INVERSE X;"D";:
  PRINT "  CLASSIFICHE"' 'P$:
  PAUSE O: IF INKEY$<>"U"
  THEN RETURN
2010 GO TO VAL "9"*W
3000 GO SUB W+W: IF INKEY$="N" THEN
  SAVE "SQUADRE" DATA A$( )
3010 IF INKEY$="D" THEN SAVE "CL"
  CODE VAL "31040",VAL "1728"
3020 GO TO W+W+W
4000 GO SUB W+W: IF INKEY$="N" THEN
  LOAD "SQUADRE" DATA A$( )
4010 IF INKEY$="D" THEN LOAD "CL"
  CODE
4020 GO TO VAL "4"*W
5000 GO SUB VAL "7"*W: FOR F=X TO R
  STEP X+X: PRINT INVERSE X;
  CHR$(CODE "@"+F);: PRINT " ";A$(
  F+K), INVERSE X;CHR$(CODE "A"+
  F);: PRINT " ";A$(F+K+X)''':
  NEXT F: PRINT AT VAL "21",VAL "
  6";P$: PAUSE O: LET J=CODE
  INKEY$+K-CODE "@": IF J<=K OR J>
  K+R THEN GO TO W
5010 GO SUB VAL "7"*W: PRINT
  INVERSE X;"N": PRINT ', "IN CAS
  A  FUORI"' ' "VINTE"' ' "PARI"' ' "PE
  RSE"' ' "FATTE"' ' "SUBITE"' ' "Pt. UL
  TIME 5 G."''P$: LET I=J*B+VAL "
  31028": LET L=I+B-X
5020 PRINT AT X,X+X;A$(J): PRINT #O;
  AT O,O;TAB VAL "31";"
  : LET G=VAL "6": FOR F=O TO
  VAL "9": PRINT AT G,VAL "17"+
  VAL "10"*(F>=VAL "5"); INVERSE X
  ;F;: PRINT " ";PEEK (I+F);" ":
  LET G=G+X+X-VAL "10"*(F=VAL "4"
  ): NEXT F: PRINT INVERSE X;AT G
  ,VAL "22";"A";: PRINT " ";PEEK (
  L-X);" ": PAUSE O: LET G=CODE
  INKEY$-CODE "0": IF G<O OR G>
  VAL "17" AND G<>VAL "30"
  THEN GO TO VAL "5"*W

```

```

5030 IF G>VAL "9" AND G<>VAL "30"
  THEN PRINT #O;AT O,O;"inserire
  i pt. delle 5 giornate":
  PAUSE O: LET G=CODE INKEY$-
  CODE "0": LET G=G*(G>=O AND G<=X
  +X): POKE L,O: POKE L-X,G:
  PRINT #O;G;: FOR F=INT PI TO O
  STEP -X: PAUSE O: LET G=CODE
  INKEY$-CODE "0": LET G=G*(G>=O
  AND G<=X+X): POKE L,PEEK L+G*
  VAL "4"↑F: POKE L-X,PEEK (L-X)+G
  : PRINT #O;"-";G;: NEXT F:
  GO TO VAL "5020"
5040 IF G=VAL "30" THEN INPUT "NOME:
  "; LINE A$(J): GO TO VAL "5020"
5050 INPUT ( FLASH X;G; FLASH O;" ins
  erire il valore esatto:");
  LINE M$: LET F=VAL M$*(CODE M$
  >=CODE "0" AND CODE M$<=CODE "9"
  ): POKE I+G,F*(F<CODE "d"):
  GO TO VAL "5020"
6000 GO SUB W: RANDOMIZE USR VAL "307
  82": GO SUB VAL "7"*W: PRINT "SE
  RIE ";S$'': FOR F=X TO R:
  INPUT (F;". "); LINE A$(K+F):
  PRINT F;". ";TAB VAL "4";A$(K+F
  ): NEXT F: PRINT #O;C$: PAUSE O:
  GO TO VAL "6"*W
7000 CLS : PRINT AT O,VAL "31";
  FLASH X;L$: RETURN
8000 GO SUB VAL "7"*W: PRINT "' ' " ;
  INVERSE X;"G";: PRINT "  GENER
  ALE IN ORDINE DI PUNTI"' ' " ;
  INVERSE X;"V";: PRINT "  PARTI
  TE VINTE"' ' " ; INVERSE X;"N";:
  PRINT "  PARTITE PAREGGIATE"' '
  " ; INVERSE X;"P";: PRINT "  P
  ARTITE PERSE"' ' " ; INVERSE X;"F
  ";: PRINT "  RETI SEGNALE"' ' "
  ; INVERSE X;"S";: PRINT "  RETI
  SUBITE"' ' " ; INVERSE X;"I";:
  PRINT "  MEDIA INGLESE"' ' " ;
  INVERSE X;"5";: PRINT "  Pt. U
  LTIME 5 GIORNATE"' 'P$
8010 PAUSE O: LET M$=INKEY$: IF M$="U
  " THEN GO TO W
8020 LET L=O: IF M$="I" THEN LET V$=
  "MEDIA ING.": LET J=VAL "172":
  GO TO VAL "8110"
8030 IF M$="5" THEN LET J=VAL "221":
  LET V$="Pt. ULTIME 5 G.":
  GO TO VAL "8110"
8040 LET J=VAL "120": CLS : GO SUB
  VAL "7"*W: PRINT "' ' " ;
  INVERSE X;"T";: PRINT "  TOTAL
  E"' ' " ; INVERSE X;"C";: PRINT "
  IN CASA"' ' " ; INVERSE X;"F";
  : PRINT "  FUORI": PAUSE O:
  LET I=X*(INKEY$="C")+(X+X)*(
  INKEY$="F")
8050 IF M$="V" THEN LET J=VAL "188"-
  VAL "51"*(I=X)+VAL "3"*(I>X):

```

```

      LET V$="VINTE"
8060 IF M$="N" THEN LET J=VAL "144"+
      VAL "3"*(I=X)+VAL "55"*(I>X):
      LET V$="PARI"
8070 IF M$="P" THEN LET J=VAL "151"+
      VAL "3"*(I=X)+VAL "58"*(I>X):
      LET V$="PERSE"
8080 IF M$="F" THEN LET J=VAL "158"+
      VAL "3"*(I=X)+VAL "55"*(I>X):
      LET V$="FATTE"

8090 IF M$="S" THEN LET J=VAL "165"+
      VAL "3"*(I=X)+VAL "52"*(I>X):
      LET V$="SUBITE"
8100 IF J=VAL "120" THEN LET J=J+3*(
      I=X)+75*(I>X): LET G=VAL "23522"
      +I*VAL "10": LET L=VAL "7":
      LET V$=" "+E$(X+I*L TO L+I*L)
      +" RETI"
8110 POKE VAL "30817",J: RANDOMIZE
      USR VAL "30803": CLS : LET I=
      VAL "23659": POKE I,X: PRINT "SE
      RIE";TAB B;V$'S$': IF L>X
      THEN PRINT AT x,B;"pt g v n
      p f s"
8120 FOR F=X TO R: LET J=PEEK (VAL "3
      0999"+F): PRINT A$(J);: IF L>X
      THEN RANDOMIZE USR G: PRINT
8130 IF L=O THEN PRINT TAB B;PEEK (
      VAL "31019"+J-K)-VAL "64"*(M$="I
      ")
8140 NEXT F: PRINT #O;AT O,O;C$:
      PAUSE O: POKE I,X+X: GO TO
      VAL "8"*W
8990 BORDER VAL "7": INK VAL "7":
      PAPER VAL "7": CLEAR VAL "30781
      ": LOAD ""CODE : LOAD ""CODE :
      POKE VAL "23658",VAL "8": DIM A
      $(144,10)
8995 INK VAL "9": LET W=VAL "1000":
      LET X=W=W: LET O=W<>W: LET B=
      VAL "12": LET C$=" premi un tast
      o per continuare ": LET E$="PART
      ITEIN CASAFUORI ": LET p$=" u
      PER USCIRE": LET N$="C1 GIR
      ONE AC1 GIRONE BC2 GIRONE AC2 GI
      RONE BC2 GIRONE CC2 GIRONE D"
9000 PAPER 7: BORDER 7: CLS : PRINT "
      MENU' GENERALE"""" "; INVERSE X
      ;"A";: PRINT " AGGIORNAMENTO
      CLASSIFICHE"""" "; INVERSE X;"C"
      ;: PRINT " CLASSIFICHE"""" ";
      INVERSE X;"L";: PRINT " CARI
      CAMENTO DATI"""" "; INVERSE X;"S
      ";: PRINT " SALVATAGGIO DATI"
      """" "; INVERSE X;"E";: PRINT "
      CORREZIONE CLASSIFICHE"""" ";
      INVERSE X;"I";: PRINT " INIZ
      IALIZZAZIONE"
9010 LET L$=INKEY$: IF L$="A" THEN
      PAPER VAL "6": BORDER VAL "6":
      GO TO W
9020 IF L$="C" THEN BORDER VAL "5":
      PAPER VAL "5": GO TO W

```

```

9030 IF L$="L" THEN PAPER VAL "2":
      BORDER VAL "2": GO TO VAL "4"*W
9040 IF L$="S" THEN BORDER VAL "3":
      PAPER VAL "3": GO TO VAL "3"*W
9050 IF L$="E" THEN PAPER VAL "4":
      BORDER VAL "4": GO TO W
9060 IF L$="I" THEN BORDER X:
      PAPER X: GO TO VAL "6"*W
9100 GO TO VAL "9010"
9999 CLEAR : SAVE "R&C CALCIO" LINE
      VAL "8990"

```

Classifiche dei campionati
versione per ZX Spectrum 16K/48K
machine code

```

5 REM routine AGGIORNAMENTO
10 DATA 33,0,0,41,41,68,77,41,9,201
,33,0,0,205,3,91,1,61,121,9,235,
33
20 DATA 0,0,205,3,91,1,56,121,9,34,
85,91,1,0,0,126,128,119,43,126,1
29,119
30 DATA 26,129,18,29,26,128,18,27,1
25,214,3,111,121,184,62,0,40,6,4
8,8,35,27
40 DATA 198,64,35,27,198,64,52,235,
52,79,125,230,252,111,121,205,92
,91,33,68,121,121
50 DATA 237,68,198,128,17,7,0,25,94
,23,203,19,23,203,19,23,115,45,2
35,6,4,38
60 DATA 0,41,41,132,16,249,235,119,
201,0
65 REM SUBroutines
70 DATA 205,195,91,205,137,91,24,6,
205,203,91,205,154,91,205,147,91
,46,64,38,121,25
80 DATA 134,201,205,199,91,46,65,24
,244,205,209,91,46,66,24,237,205
,213,91,46,67,24
90 DATA 230,205,217,91,46,68,24,233
,205,154,91,23,205,209,91,205,14
7,91,237,68,198,64
100 DATA 24,3,205,137,91,46,69,24,20
0,205,191,91,23,46,70,24,192,205
,199,91,205,191
110 DATA 91,46,71,24,182,46,72,24,17
8,46,73,24,174,46,74,24,170,0
115 REM routines STAMPA
120 DATA 205,178,120,120,128,188,144
,151,158,165,205,178,120,123,131
,137,147,154,161,168,205,178
130 DATA 120,195,203,191,199,209,213
,217
150 RESTORE 10: FOR F=23296 TO 23296
+255: READ A: POKE F,A: NEXT F
195 REM routine AZZERAMENTO
200 DATA 42,136,120,205,3,91,125,205
,0,91,17,64,121,25,71,175,119,35
,16,252,201

```

```

205 REM routine ORDINAMENTO
210 DATA 205,0,91,58,136,120,71,221,
    33,44,121,235,175,205,128,91,33,
    12,0,25,221,119
220 DATA 0,221,35,16,240,6,20,33,24,
    121,175,119,44,16,252,17,44,121,
    58,136,120,50
230 DATA 160,120,71,197,33,44,121,1,
    0,16,26,190,48,1,12,44,16,249,28,
    33,23,121
240 DATA 9,175,44,190,32,252,193,58,
    1,91,198,16,144,60,119,16,220,62,
    24,50,188,120
250 DATA 33,108,64,34,181,120,201
255 REM SUBroutine II
260 DATA 209,213,33,108,80,6,7,229,1
    97,42,40,121,45,38,0,205,3,91,19
    3,235,126,111
270 DATA 38,91,229,33,210,120,227,22
    9,175,201,30,255,14,2,225,254,10
    48,4,13,44,24
280 DATA 9,214,10,28,48,251,198,10,8
    123,198,48,22,30,23,23,23,203,1
    8,95,197,6
290 DATA 8,76,26,119,28,36,16,250,44
    40,1,97,193,8,13,32,227,44,209,
    28,213,16
300 DATA 174,125,198,11,111,34,181,1
    20,33,188,120,52,209,201
350 RESTORE 200: FOR F=30782 TO 3099
    9: READ A: POKE F,A: NEXT F
400 SAVE "23296"CODE 23296,256:
    SAVE "30782"CODE 30782,218

```



Novità firmate Jackson.

Le Novità Jackson puoi acquistarle in via Mascheroni 14 a Milano e in tutte le migliori librerie.



GRUPPO EDITORIALE JACKSON

La biblioteca che fa testo.

Nursery

di Luciano Prelli

Un simpatico arcade dal tema originale

Lo scopo del gioco è quello di aiutare un'infermiera ad allattare i dieci neonati piangenti presenti nella "nursery". Questa è costituita da due stanze: a) la cucina, col pavimento blu, in cui si trovano, dall'alto verso il basso: 12 biberon, il frigorifero, i fornelli, il lavello; b) la sala, con pavimento a quadratoni, in cui si trovano i 10 neonati.

In alto sono visibili: su fondo rosso il numero di neonati ancora da allattare; su fondo nero il numero di biberon a disposizione. In basso sono evidenziate le infermiere utilizzabili (oltre a quella "in campo") e nel riquadro sottostante i punti e il tempo.

Per raggiungere lo scopo (cioè allattare tutti i 10 neonati) bisogna, oltre che spostare la "nurse" con i soliti tasti del cursore, svolgere alcune "operazioni" specifiche. Queste consistono nel:

- 1) prendere il biberon;
- 2) lavarlo;
- 3) riempirlo di latte;
- 4) scaldarlo.

La prima operazione si esegue ponendosi al di sotto del biberon prescelto e premendo il tasto "B": in alto il numero di biberon diminuirà di uno.

Quindi bisogna portarsi in basso, nell'angolo più a sinistra, vicino al lavello: la pressione del tasto "L" fa sì che il biberon venga accuratamente lavato (!). Successivamente occorre guidare l'infermiera verso il frigorifero, posizionarla esattamente davanti ad esso e prelevare il latte (tasto "M", dall'inglese "milk").

Infine, per scaldare il latte, bisogna portare la nurse vicino al fornello (2/3 superiori di esso) e premere il tasto "S" più volte. Il grado di riscaldamento è indicato dal colore assunto dall'infermiera (dal giallo al magenta): perché il neonato smetta di piangere il latte deve essere almeno "verde".

Una volta che la vostra infermiera si è ben ben colorata è necessario uscire dalla cucina, cioè oltrepassare la porta ("grate" gialle) che si chiude e apre



Descrizione del programma

Linee	Commento
10-25 30	presentazione richiama le subroutine di definizione caratteri e preparazione dello schermo
40-110	inizializzazione variabili
120	stampa l'infermiera nella posizione (y,x)
130	salta alla routine di stampa della porta
140	memorizza la posizione attuale dell'infermiera
220-240	aggiornano l'orologio; saltano alla fine, se il tempo è scaduto (time 1=0)
2502	aggiorna il punteggio
270-340	lettura tastiera e salto alla routine opportuna
400-500	preleva il biberon
550-600	flava il biberon
650-700	prelevano il latte
750-800	richiamano la routine di riscaldamento del latte (2200)
850-900	se la posizione è esatta saltano alla routine di allattamento (2500)
1000-1090	routine di movimento
2000-2130	routine di stampa casuale della porta

automaticamente (ma a caso). Attenzione: se la porta è chiusa, non si può passare, ma se si chiude mentre state passando perdete un'infermiera con relativo biberon (avete a disposizione 5 infermiere e 12 biberon). Oltrepassata la porta bisogna portarsi a fianco di uno dei lettini dei neonati ancora piangenti (UEEH), nei 2/3 superiori di esso, sul lato destro, e allattare premendo il tasto "A".

L'infermiera tornerà di colore bianco e potrete ricondurla in cucina (attenti alla porta) per ricominciare.

Il tutto va ripetuto per 10 volte (quanti sono i neonati), nel breve tempo di circa 6 minuti (360 "secondi"): se vi sembra poco, potete assegnare alla variabile "time" un valore più alto (il massimo è 1020 con l'orologio così com'è) oppure modificare il contatempo.

I punteggi ottenibili sono:

- 1) biberon = 5 punti;
- 2) lavaggio = 10 punti;
- 3) latte = 15 punti;
- 4) riscaldamento = 5 punti per ogni grado (max 20 punti);
- 5) allattamento = 50 punti;
- 6) la porta, chiudendosi sull'infermiera, toglie 100 punti;
- 7) se riuscite ad allattare tutti i neonati prima dello scadere del tempo, ottenete 10 punti per ogni secondo risparmiato;
- 8) se totalizzate più di 250 punti, riceve altri 100 punti per ogni infermiera risparmiata.

Particolarità del programma

1) linea 120: stampa i caratteri grafici relativi all'infermiera, a seconda del valore di due variabili:

a) "d" = direzione del movimento (0=sinistra; 1=destra). Viene modificata in linea 1040, a seconda del tasto premuto: notare l'uso degli operatori logici. Serve a stampare l'infermiera correttamente girata verso destra o verso sinistra.

b) "car" = posizione delle gambe (0 = verticali; 1=oblique). Viene modificata in linea 1010 (car = NOT car : se car = 1 diventa car = 0 e viceversa). Serve a simulare la camminata dell'infermiera.

Nursery

versione per ZX Spectrum 16K/48K

```

10 REM *****
11 REM *
12 REM * NURSERY *
13 REM *
14 REM * by Luciano Prelli *
15 REM *
16 REM * TRIESTE *
17 REM *
18 REM *****
19 REM

```

2100

se l'infermiera passa mentre la porta si chiude, aggiorna il punteggio e la situazione

2200-2230

routine di riscaldamento del latte

2500-2590

routine di allattamento dei neonati. Se sono stati allattati tutti, salta alla 2800

2800

segnala che tutti i neonati sono stati allattati e aggiorna il punteggio

3000-3110

fine del gioco

4000-4470

preparano lo schermo

5000-5050

preparano i caratteri grafici

6000-6190

DATA per i caratteri grafici e la musicchetta

7000-7010

SAVE del programma

Descrizione delle variabili

time	contiene il tempo massimo per il gioco
time1	è il valore del tempo che viene calcolato e stampato
pt	punteggio
nb	numero di biberon a disposizione
na	numero di neonati ancora da allattare
p	variabile di controllo per la porta (1 = aperta; 0 = chiusa)
y	linea a cui stampare la parte superiore dell'infermiera
x	colonna a cui stampare la parte superiore dell'infermiera
car	variabile di controllo per il movimento delle gambe
d	variabile di controllo per l'orientamento dell'infermiera
bi	assume il valore 1, se l'infermiera porta il biberon
la	assume il valore 1, quando il biberon viene lavato
mi	assume il valore 1, se è stato prelevato il latte
sc	assume il valore 1, se il latte è stato scaldato
co	contiene il colore di INK per la stampa dell'infermiera
yl	colonna a cui stampare uno spazio al posto dell'infermiera durante il movimento
xl	linea a cui stampare uno spazio al posto dell'infermiera durante il movimento
i\$	contiene una stringa corrispondente al tasto premuto
n,m	variabili di controllo per vari cicli FOR...NEXT
a,b,c,d	usate nella preparazione dello schermo e lettura dei DATA

2) linea 220: orologio funzionante sulla variabile di sistema FRAMES posta nelle locazioni 23672/23673.

3) Funzione SCREEN\$: ritorna, con argomento (x,y), una stringa contenente il carattere presente nella posizione (x,y). Nel nostro caso l'infermiera può muoversi solo se la prossima posizione è uno spazio (" ").

Le linee tracciate coi comandi grafici (PLOT, DRAW, CIRCLE) ed i caratteri grafici non vengono riconosciuti: il risultato è una stringa vuota (""), diversa

quindi da uno spazio, per cui l'infermiera non può sovrapporsi a queste figure. Nel caso del riconoscimento della porta chiusa è ovvio che "#" è diverso da " ".

4) Funzione ATTR: ritorna, con argomento (y,x), il valore degli attributi nella posizione (y,x). È usata per evitare all'infermiera di camminare sui muri (ATTR = 41) e per riconoscere i neonati (ATTR = 7 (bianco su nero) o ATTR = 23 (bianco su rosso)).

```

20 BORDER 1: PAPER 1: INK 7: CLS
21 PRINT AT 8,10;"N U R S E R Y"
22 PRINT AT 10,13;"© 1984"
23 PRINT AT 12,8;"by Luciano Prelli
"
25 PRINT AT 18,9; FLASH 1;"Attender
e prego"
30 GO SUB 5000: GO SUB 4000
40 LET time=360: LET pt=0: LET nb=1
2: LET na=10: LET ni=28
50 FOR n=0 TO 3: PRINT AT 19,25+n;"
N";AT 20,25+n;"O": NEXT n
60 PRINT AT 1,21; PAPER 2;na;AT 1,2
6; PAPER 0;nb
70 REM Routine principale

```



```

80 LET p=0: LET x=26: LET y=14:
   LET car=0: LET d=0
90 LET bi=0: LET la=0: LET mi=0:
   LET sc=0
100 LET co=7
110 PRINT INK 6; PAPER 8; AT 8,24;"#
   "; AT 9,24;"#"
120 PRINT PAPER 8; INK co; AT y,x;
   CHR$(154+3*d); AT y+1,x; CHR$(15
   5+3*d+car)
130 GO TO 2000
140 LET xl=x: LET yl=y
220 LET timel=time-4*PEEK 23673-
   INT ((PEEK 23672)/64)
230 IF timel<=0 THEN LET timel=0:
   PRINT #1; AT 0,24; FLASH 1;"FINI
   TO": GO TO 3000
240 PRINT #1; AT 0,24; timel;" "
250 PRINT #1; AT 0,8; pt;" "
270 LET i$=INKEY$
280 IF i$="b" THEN GO TO 400
290 IF i$="l" THEN GO TO 550
300 IF i$="m" THEN GO TO 650
310 IF i$="s" THEN GO TO 750
320 IF i$="a" THEN GO TO 850
330 IF i$>="5" AND i$<="8" THEN
   GO SUB 1000
340 GO TO 120
400 IF bi=0 AND y<=6 AND SCREEN$(y-
   1,x)<>" " THEN PRINT PAPER 8
   ; AT y-1,x;" ": BEEP .005,24:
   LET bi=1: LET nb=nb-1: LET pt=pt
   +5: PRINT PAPER 0; AT 1,26;" "
   ; AT 1,26; nb
450 IF nb=0 THEN PRINT FLASH 1;
   PAPER 2; AT 1,25;"FINE"
500 GO TO 150
550 IF la=0 AND y=14 AND x=25 AND bi
   =1 THEN LET la=1: LET pt=pt+10:
   BEEP .005,24
600 GO TO 150
650 IF mi=0 AND y=8 AND x=26 AND bi=
   1 AND la=1 THEN LET mi=1: LET pt
   =pt+15: BEEP .005,24
700 GO TO 150
750 IF y=12 AND x=26 AND bi=1 AND la
   =1 AND mi=1 THEN LET sc=1:
   BEEP .005,24: GO SUB 2200
800 GO TO 120
850 IF x<23 AND (y=4 OR y=7 OR y=11
   OR y=14) AND (ATTR (y,x-1)=7
   OR ATTR (y,x-1)=23) AND co<5
   THEN GO TO 2500
900 GO TO 150
1000 REM Movimento
1010 LET car=NOT car
1020 BEEP .02,-25
1030 LET x=x+(1 AND i$="8")-(1 AND i$
   ="5")
1040 LET d=(1 AND i$="8")+(0 AND i$="
   5")+(d AND (i$="6" OR i$="7"))
1050 LET y=y+(1 AND i$="6")-(1 AND i$
   ="7")
1060 PRINT PAPER 8; AT yl,xl;" "; AT y
   l+1,xl;" "
1070 IF SCREEN$(y,x)<>" " OR
   ATTR (y,x)=41 THEN LET x=x+1:
   LET y=yl
1080 IF SCREEN$(y+1,x)<>" " OR
   ATTR (y+1,x)=41 THEN LET x=x+1:
   LET y=yl
1090 RETURN
2000 REM Porta
2010 LET p=INT (RND*3)
2020 IF p THEN PRINT PAPER 8; AT 8,2
   4;" "; AT 9,24;" ": BEEP .002,-20
2030 IF NOT p THEN IF SCREEN$(8,24)
   ="" THEN GO TO 2100
2040 IF NOT p THEN PRINT ; INK 6;
   PAPER 8; AT 8,24;"#"; AT 9,24;"#
   ": BEEP .003,-20
2050 GO TO 140
2100 FOR n=30 TO 0 STEP -1: BEEP .002
   ,n: BEEP .002,n-10: NEXT n
2110 LET pt=pt-100: IF ni=24 THEN
   GO TO 3000
2115 PRINT AT 19,ni;" "; AT 20,ni;" ":
   LET ni=ni-1
2130 GO TO 80
2200 REM Riscalda
2210 FOR n=0 TO 50: NEXT n
2220 IF co=3 THEN RETURN
2230 LET co=co-1: BEEP .1,12-2*co:
   LET pt=pt+5
2240 RETURN
2500 REM Allatta
2510 PRINT PAPER 8; AT y-1,x-2;" "
2530 FOR n=1 TO 50: BEEP .004,n:
   BEEP .002,n-5: LET pt=pt+1:
   PRINT #1; AT 0,8; pt: NEXT n
2550 LET na=na-1
2560 PRINT AT 1,21; PAPER 2; na;" "
2570 IF na=0 THEN GO TO 2800
2580 IF nb=0 THEN GO TO 3000
2590 GO TO 90
2800 REM Tutti allattati
2810 PRINT #1; AT 0,15; FLASH 1;"BRAVI
   SSIMO !!!"
2820 LET pt=pt+10*timel
3000 REM Fine
3010 LET pt=pt+100*((ni-24) AND (pt>2
   50))
3020 PRINT #1; AT 0,8; pt;" "
3030 RESTORE 6180: FOR n=0 TO 12
3040 READ a,b
3050 BEEP a,b
3060 NEXT n
3070 PRINT PAPER 8; AT y,x;" "; AT y+1
   ,x;" "
3080 PRINT #1; AT 0,15;" DI NUOVO?
   "
3090 PAUSE 1: PAUSE 0: IF INKEY$=""
   THEN GO TO 3090
3100 IF INKEY$="s" THEN GO SUB 4330:
   GO TO 40

```

```

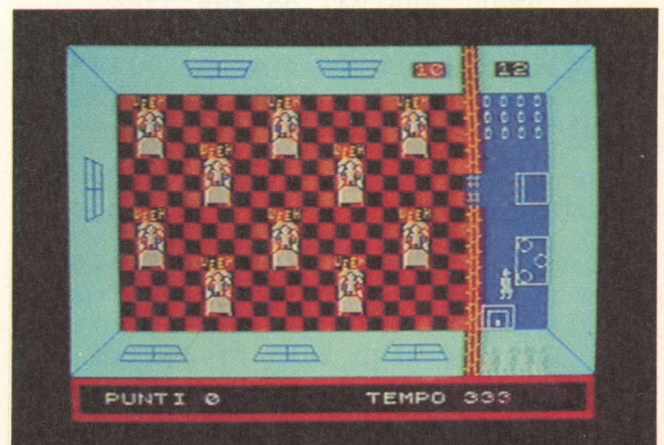
3110 STOP
4000 REM Schermo
4010 BORDER 0: PAPER 5: INK 1: CLS :
    LET a=1
4020 FOR n=3 TO 17
4030 FOR m=3 TO 23
4040 PRINT AT n,m; PAPER 2*a;" ":
    LET a=NOT a
4050 PRINT AT n,24; PAPER 1;" "
4060 NEXT m: NEXT n
4070 FOR n=1 TO 4
4080 READ a: READ b: READ c: READ d
4090 PLOT a,b: DRAW c,d
4100 NEXT n
4110 FOR n=0 TO 20
4120 PRINT AT n,24; PAPER 6; INK 2;"A
"
4130 IF n=7 THEN LET n=9
4140 NEXT n
4150 INK 1
4160 FOR n=24 TO 152 STEP 64
4170 PLOT n,16: DRAW 32,0: DRAW -4,8:
    DRAW -24,0: DRAW -4,-8: PLOT n+
    2,20: DRAW 28,0: PLOT n+16,16:
    DRAW 0,8
4180 NEXT n
4190 PLOT 8,88: DRAW 0,32: DRAW 8,-4:
    DRAW 0,-24: DRAW -8,-4: PLOT 8,
    104: DRAW 8,0: PLOT 12,90:
    DRAW 0,28
4200 FOR n=56 TO 120 STEP 64
4210 PLOT n,168: DRAW 32,0: DRAW -4,-
    8: DRAW -24,0: DRAW -4,8: PLOT n
    +2,164: DRAW 28,0: PLOT n+16,168
    : DRAW 0,-8
4220 NEXT n
4230 INK 7: PLOT 231,79: DRAW -15,0:
    DRAW 0,-24: DRAW 15,0: DRAW 0,2
    4
4240 CIRCLE 220,75,3
4250 CIRCLE 228,67,3
4260 CIRCLE 220,59,3
4270 PLOT 215,32: DRAW 0,12: DRAW -15
    ,0: DRAW 0,-12: DRAW 15,0
4280 PLOT 212,32: DRAW 0,9: DRAW -9,0
    : DRAW 0,-9
4290 CIRCLE 207,35,1
4300 PLOT 231,111: DRAW -15,0: DRAW 0
    ,-15: DRAW 15,0: DRAW 0,15
4310 PLOT 219,111: DRAW 0,-15
4320 NEXT n
4330 FOR n=25 TO 28
4340 FOR m=3 TO 5
4350 PRINT AT m,n; PAPER 8;"B"
4360 NEXT m: NEXT n
4370 LET a=0
4380 FOR m=4 TO 20 STEP 4
4390 FOR n=4 TO 11 STEP 7
4400 PRINT PAPER 8; INK 7; AT n+3*a,m
    ; "CD"; AT n+1+3*a,m; "EF"; AT n+2+3
    *a,m; "GH"; INK 6; AT n-1+3*a,m; "I
    J"
4410 NEXT n

```

```

4420 LET a=NOT a: NEXT m
4430 PRINT AT 21,0; PAPER 0; INK 2;"
    {SG4}{30G3}{G7}"
4440 PRINT #1; AT 0,0; PAPER 0; INK 2;
    "{SG5}"; INK 7;" PUNTI 0 ";
    TAB 18;"TEMPO 240 "; INK 2; TAB 3
    1;"{G5}"
4450 PRINT #1; PAPER 0; INK 2;"{SG1}
    {30SG3}{SG2}"
4460 POKE 23673,0: POKE 23672,0
4470 RETURN
4480 RETURN
5000 REM Caratteri grafici
5010 FOR n=0 TO 127
5020 READ a
5030 POKE USR "a"+n,a
5040 NEXT n
5050 RETURN
6000 REM Dati
6010 DATA 219,219,195,219,27,219,219,
    216
6020 DATA 0,16,56,40,40,40,56,0
6030 DATA 15,30,61,35,27,57,55,43
6040 DATA 240,120,188,196,216,156,236
    ,212
6050 DATA 43,43,35,58,58,48,47,31
6060 DATA 212,212,196,92,92,12,244,24
    8
6070 DATA 63,63,63,63,63,63,32,32
6080 DATA 252,252,252,252,252,252,4,4
6090 DATA 0,160,160,174,168,236,8,14
6100 DATA 0,10,234,142,202,138,224,0
6110 DATA 16,56,60,56,16,124,124,124
6120 DATA 124,60,60,126,126,36,36,108
6130 DATA 124,60,60,126,126,36,66,198
6140 DATA 8,28,60,28,8,62,62,62
6150 DATA 62,60,60,126,126,36,36,54
6160 DATA 62,60,60,126,126,36,66,99
6170 DATA 0,8,24,24,255,8,-24,24,255,
    175,-24,-24,0,175,24,-24
6180 DATA .3,0,.1,0,.4,3,.4,0,.4,0,.8
    ,3,.3,0,.1,3,.4,8,.6,7
6190 DATA .2,5,.4,5,.6,3
7000 REM Save
7010 SAVE "NURSERY" LINE 1

```



I libri italiani vengono offerti con il 10% di sconto sul prezzo di copertina

di **Lucio Bragagnolo**

Giochi con il computer - come funzionano, come si vince

di **Ian Graham**
a cura di **Lisa Watts**
ed. **Gruppo Editoriale Jackson**



Il secondo rappresentante dei libri "Speedy Computer" qui recensito ha un'attinenza solo indiretta col mondo Sinclair, diremmo più che altro delle affinità: gli autori si incaricano questa volta di accompagnare il lettore in un lungo viaggio nel variegato ed eterogeneo mondo del gioco elettronico, campo - come si può ben capire - imparentato con l'informatica, ma non in modo eccessivamente profondo. Il pianeta gioco viene esaminato in tutte le sue forme, dai giochi elettronici tascabili alle macchine da bar, alle console specializzate, ai personal computer. In realtà, nonostante le differenze esteriori e di classificazione, il cuore di tutti i marchingegni citati non è altro che un particolare computer, a volte specializzato, a volte

piccolissimo ed elementare, a volte tutotfare o incredibilmente potente. Il gioco diviene così il pretesto per un interessante viaggio nell'hardware della macchina, allo scopo di migliorare la comprensione del funzionamento interno della macchina, a partire dai concetti di RAM e ROM o CPU fino ad arrivare a una spiegazione sommaria, ma per un non esperto più che sufficiente, di cosa siano i cristalli liquidi, o di come funzioni un joystick, o di cosa sia un adventure game. Non manca neppure una trattazione sommaria, ma interessante, dei vari filoni in cui si suddividono i giochi elettronici (avventure - educativi - arcade - simulazioni...), né di cosa sarà e come funzionerà un ipotetico "gioco del 2000".

I patiti ad oltranza dei cosiddetti "coin - op" da bar potranno poi trovare, sparse qua e là tra le pagine di "Giochi con il computer", tutta una serie di curiosità e consigli sui grandi miti di questa forma di divertimento, come "Pacman", "QIX", "Frogger", "Battlezone", "Galaxian" e via dicendo. Sapevate ad esempio che "Asteroids" può visualizzare sullo schermo non più di 35 oggetti contemporaneamente, e che questa limitazione può portare sensibili vantaggi (provate a pensare perché) a un giocatore esperto?

Non manca nemmeno, a completare questo sintetico, ma esaustivo, trattato sul gioco, un capitolo dedicato alla storia dei giochi elettronici, dove gli interessati potranno scoprire, altro esempio, che il primo videogioco inventato non è, come molti pensano, il celebre "Pong", che imperversava fino a qualche anno fa sui televisori di ogni parte d'Italia...; merita una menzione, infine, una sezione discretamente ampia dedicata ai giochi intelligenti e in particolare modo agli scacchi elettronici.

Per riassumere, "Giochi con il computer" si propone come una via di mezzo tra l'intrattenimento e l'informazione, servendosi di un argomento tutt'altro che serio, ma trattandolo non certo futilmente. Il suo contenuto informativo non è davvero quello di un trattato

scientifico, né la sua lettura aiuterà a sviluppare brillantissimi giochi d'avventura o di azione, magari scritti in linguaggio macchina: crediamo, però, che questo libro possa costituire una divertente e interessante pausa di rilassamento, davvero quello che ogni tanto occorre per combattere il famoso logorio della vita - non solo informatica - moderna.

Primi passi in BASIC - una facile guida per scrivere programmi

di **Brian Reffin Smith**
e **Lisa Watts**
ed. **Gruppo Editoriale Jackson**



Prendiamo in esame nel corso di questa puntata di PRINT due libri di una nuova collana Jackson, intitolata "Speedy Computer". Come già si intuisce dal suo titolo, i libri

in essa compresi hanno il compito di avvicinare il lettore ai computer "velocemente", con spigliatezza e senza annoiare con lunghi tomi di considerazioni spesso solo teoriche, a patto però di conservare la propria efficacia dal punto di vista didattico.

L'approccio col lettore vuole essere così il più amichevole possibile: copertina coloratissima e più simile a quella di un libro per bambini piuttosto che a un testo di informatica, numerosi "dialoghi" sintetici e densi di informazioni, vignette esplicative, grande colloquialità nel linguaggio: ecco le armi quasi più di seduzione che di convincimento usate dai libri "Speedy Computer".

Più dettagliatamente "Primi passi in BASIC" si propone di far esplorare un po' alla volta al lettore il linguaggio più usato dai personal/home computer, esaminandone gradatamente le caratteristiche principali. Ogni punto delle spiegazioni è corredato da un breve esempio scritto in un BASIC il più possibile standard: dove le modifiche erano inevitabili, come nel caso della grafica o della gestione delle stringhe, al termine del libro sono stati riportati i cambiamenti da eseguire. Il libro è stato realizzato in BASIC Apple; bisogna dire tuttavia che vi è ben poco da cambiare, dato che la maggioranza dei listati girerà anche sul nostro amato ZX senza difficoltà; in più l'acquisizione di qualche informazione sugli altri BASIC non fa certo male, e anzi può servire a valutare con maggior cognizione di causa quali siano i sistemi da preferire (e consigliare agli amici...), oltre che migliorare il proprio bagaglio di cultura informatica generale.

Proseguendo nelle spiegazioni, gli esempi divengono progressivamente sempre più complessi, fino a fornire listati come quelli di un piccolo data base o di alcuni metodi di ordinamento di stringhe, per finire con alcuni simpatici giochi privi, intendiamoci, di effetti grafici e sonori (altrimenti il numero delle modifiche da eseguire salirebbe alle stelle), ma ugualmente godibili dai principianti e utili per chi si accosta al libro per imparare. Non c'è molto altro da dire, anche perché l'efficacia di questi libriccini (solo in termini di volume) è data dalla valida mescolanza di testo e grafica, cosa questa difficile da evidenziare in una recensione solo scritta. Il modo migliore di convincersi della validità degli "Speedy Computer" è quello di entrare in una libreria e aprirne qualche pagina a caso, dopo avere compiuto la stessa operazione su un testo convenzionale: invece di lunghe e noiose trattazioni, cominciate pagine e pagine prima e che termineranno dopo altrettanti fogli, si troveranno spiegazioni chiare, concise e precise, corredate da esempi concreti di rapidissima fruibilità, il tutto con un aspetto accattivante, che diventa familiare dopo la prima occhiata. Provare per credere...e per rimanere conquistati.

Imparare a programmare con lo ZX Spectrum

di **Rita Bonelli e Marco Spada**
ed. **Gruppo Editoriale Jackson**

Rita Bonelli, apprezzata autrice di numerosissimi testi dedicati un po' a tutta la famiglia degli home e personal computer, si ripresenta alla nostra attenzione con un altro ottimo libro riguardante lo Spectrum, diretto agli aspiranti programmatori. Questa volta l'approccio è però piuttosto differente: mentre nelle precedenti opere infatti — ci spiega l'autrice stessa nella sua prefazione —, pur rimanendo l'intento di introdurre alla programmazione, l'impostazione generale del libro si basava sull'apprendimento di uno specifico linguaggio di programmazione, questa volta l'idea di fondo è di concentrare l'attenzione sulla realizzazione di un programma, partendo dalla sua prima concezione per approdare alla stesura finale sul computer. Anche la scelta del linguaggio di programmazione utilizzato è in parte nuova, avendo l'autrice scelto di lavorare non solo in BASIC, ma anche in Assembler, usando per l'occasione l'abbastanza famoso editor della Picturesque, peraltro uno dei migliori sul mercato. La scelta del linguaggio macchina non sconcerta comunque i potenziali lettori: chi vuole potrà avvalersene per superare le lungaggini e gli impicci del BASIC, e chi non è interessato all'argomento sarà contento di sapere che tutte le routine sono state debitamente tradotte in linee DATA, ricopiabili senza problemi col più lento, ma amichevole interprete dello Spectrum. Qualche novità dunque, anche se lo stile rimane quello a cui siamo stati abituati dalla Bonelli: solido, logico, pragmatico, qualche volta un po' sbrigativo, ma sempre efficace e chiaro. Dicevamo dell'impostazione di fondo basata sulla realizzazione di un programma: dopo la prefazione e due capitoli introduttivi, incaricati di chiarire agli incompetenti gli aspetti fondamentali dei computer in generale e della programmazione in particolare, si entra infatti subito in argomento, presentando in ogni capitolo uno o più problemi teorici da "risolvere", concretizzandoli in un programma BASIC o Assembler attraverso tutti i passi necessari al raggiungimento dell'obiettivo. Ci sentiamo a questo punto di fare una piccola critica, rilevando come alcuni dei temi scelti (come il calcolo

del valore di un polinomio) siano di non immediata comprensione per il lettore medio, anche se, grazie all'impostazione del libro, non è tanto importante la comprensione del problema quanto l'apprendimento del metodo necessario per risolverlo nel modo migliore. Ogni argomento viene comunque prima sviscerato teoricamente, per poi essere man mano trasformato in un programma completo, che verrà infine presentato in forma definitiva. Durante tutto il percorso possiamo vedere una grande abbondanza di commenti e aiuti diretti a facilitare la comprensione del lettore. Per quanto riguarda l'output visivo notiamo la realizzazione dei testi, di buona estetica e leggibilità, ottenuta presumibilmente con una stampante a margherita, mentre per i listati si è ricorsi a una classica Seikosha GP50S con nastro nuovo e ben oliata per l'occasione. Non mancano infine tabelle un po' di tutti i generi, nitide e ben realizzate.

Il tutto occupa circa poco meno della metà del libro; il resto delle pagine ospita una vasta sezione di...appendici, le quali però, lungi dall'essere inutili pro-



lungamenti inseriti tanto per fare volume, a un esame anche solo superficiale rivelano di meritare il grande spazio ad esse destinato. La prima tratta di tutte le caratteristiche del BASIC Spectrum: precisione numerica, priorità delle operazioni (chi sa se lo Spectrum esegue prima il calcolo di una FN o uno "slicing"?), operatori logici, i cosiddetti

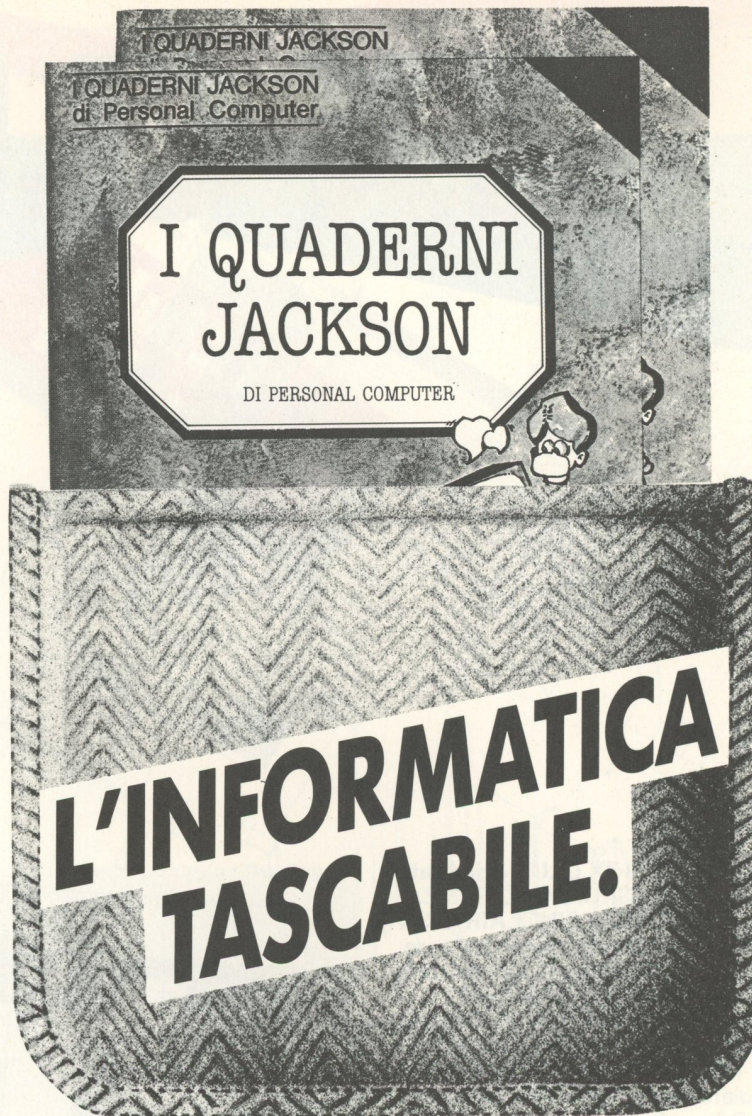
“stream” usati per lo più nelle operazioni connesse all’uso di microdrive e interfaccia 1, più una descrizione ricca ed esaustiva di tutte le parole chiave del BASIC Spectrum.

Dato il gemellaggio che viene fatto nel libro tra BASIC e Assembler, non poteva mancare una sezione riservata a quest’ultimo: è l’appendice B, contenente una sintetica descrizione dei modi di indirizzamento dello Z80, insieme all’elenco di tutte le istruzioni Assembler; è presente anche una breve trattazione di alcune peculiarità del programma della Picturesque utilizzato, come già detto, dall’autrice.

La terza appendice, molto rara a trovarsi in analoghe pubblicazioni, è estremamente preziosa: contiene difatti una serie di utilissimi consigli sull’utilizzo ottimale della memoria dello Spectrum, una dettagliata documentazione su tutte le variabili di sistema – comprese quelle relative ai rapporti tra ZX e interfaccia 1 – e varie informazioni sulla gestione dello stack, il tutto condito dall’indicazione di vari trucchi tra i più comunemente usati per risparmiare memoria. L’appendice D si intitola “Caratteri e codici del sistema” e si occupa del set di caratteri ASCII, distinguendo tra caratteri stampabili, caratteri di controllo (gli osservatori più acuti scopriranno la presenza di un carattere usato per proteggere i programmi registrati su microdrive) e token, cioè le parole chiave contenute nel set ASCII dello Spectrum a partire dal codice 165, corrispondente a RND. Questa è anche l’ultima appendice importante: seguono ancora brevi sezioni dedicate a stampanti, uso della tastiera e interfaccia 1, di cui la prima francamente piuttosto inutile e le altre tutto sommato nella normalità.

“Imparare a programmare con lo ZX Spectrum” è un libro decisamente ben realizzato. Il suo unico (veniale) difetto è di indulgere a volte eccessivamente verso un tono troppo asettico, quasi accademico, ma sa farsi perdonare adeguatamente, grazie al grande quantitativo di informazioni preziose in esso contenuto, certamente di valido aiuto per i programmatori di ogni livello e capacità.

I libri possono essere ordinati utilizzando il coupon pubblicato a pag. 82



Quaderni Jackson:
l’informatica a tutti i livelli, in una collana aperta, pratica, essenziale, aggiornata.

Tutto quello che è importante sapere sui computer, la programmazione, i linguaggi, il software, le applicazioni e i nuovi sviluppi dell’informatica.

Ogni mese, 2 volumi.

Volumi già pubblicati:

- Gianni Giaccaglioni*
“Vivere col Personal Computer”
- Paolo Bozzola*
“Dentro e fuori la scatola”

- Enrico Odetti*
“Ed è subito BASIC Vol. I”
“Ed è subito BASIC Vol. II”
- Paolo Capobussi*
e *Marco Giacobazzi*
“A ciascuno il suo Personal”
- Fulvio Francesconi*
e *Fernando Paterlini*
“To do or not to do”
- Gianni Giaccaglioni*
“Strutturare il software”
- Enrico Odetti*
“Dizionario informaticese”

**In edicola,
a sole lire 6.000.**



GRUPPO EDITORIALE JACKSON

SAN FRANCISCO - LONDRA - MILANO

ZX CLUB

I club
Sinclair

Questa puntata di ZX CLUB è dedicata interamente al GRUPPO UTILIZZATORI COMPUTER SINCLAIR/MSX NAPOLI. Ricordate il viaggio intrapreso in Inghilterra da alcuni soci del club in visita agli stabilimenti Sinclair? Orbene, Emilio TRIUNFO ed Edgardo BELLINI ci presentano un articolo sulla storia della Sinclair, e di Sir Clive in particolare, frutto delle loro peregrinazioni. Prima di lasciare la parola agli autori (vedrete che il materiale è tanto e interessante), ecco l'indirizzo del club: Gruppo Utilizzatori Computer Sinclair MSX/Napoli
c/o Roberto Chimenti
via Luigi Rizzo 18
80124 NAPOLI NA
(Tel. 081/7623121)

Sinclair: oltre il sipario

di **Emilio Trinfuo**
e **Edgardo Bellini**

Da questo numero prende inizio una rubrica che intende far conoscere – come suggerisce il titolo – i retroscena meno noti del favoloso mondo Sinclair. Partiremo, quindi, con la biografia del nostro “padre spirituale”, Sir Clive Sinclair, per proseguire con la descrizione di una visita alla Sinclair Research di Cambridge, nonché alcune interessanti interviste ad alcuni componenti e progettisti dello staff della Sinclair, ed ulteriori articoli che vi introdurremo di volta in volta.



PRIMA PARTE Clive Sinclair compare sulla scena: 1962-1979

Clive Sinclair nacque a Londra nel 1940. Si diplomò al St. George College di Weybridge all'età di 17 anni; per quattro anni lavorò per diverse riviste, in particolare di scienza e di tecnica. Nel 1962 fondò a Islington, nei pressi di Londra, una società di cui fu il maggior azionista: la *Sinclair Radionics*. I primi prodotti di questa piccola azienda furono kit di radio ed amplificatori, la cui vendita avveniva tramite ordinazione per posta. Da questo momento la Sinclair si impose come pioniere nel

campo dell'elettronica da consumo, in particolar modo nell'elettronica miniaturizzata (non dimentichiamo che negli anni '60 le radioline tascabili erano considerate prodotti di alta tecnologia, come oggi si considerano i computer). La diffusione dei prodotti dell'elettronica ebbe un tale impulso che nel 1967 gli utili della società superarono il miliardo di lire (qui e più innanzi ci riferiremo sempre alla valuta italiana corrente), e la gamma dei prodotti includeva anche sistemi HI-FI. Mr. Sinclair decise allora di trasferirsi a Cambridge e più tardi, nel 1972, data la continua espansione della società, a St.Ives.

Nel 1973 la ormai famosa *Sinclair Radionics* lanciò l'*Executive*, il primo vero calcolatore tascabile del mondo. Costava poco più di 100.000 lire, un prezzo veramente irrisorio per quei tempi in cui i circuiti integrati erano una cosa poco nota! Il suo successo fu quasi inaspettato: il progetto *Executive* vinse numerosi premi e la vendita del pocket-calculator fruttò, sul solo mercato estero, entrate superiori ai 20 miliardi di lire. Questo enorme successo commerciale fece sì che la Sinclair divenisse la Numero Uno del mercato inglese delle calcolatrici.

Dal 1973 la Sinclair, sempre sotto la saggia direzione di Mr. Sinclair, spinta dalla prospettiva di un mercato di elettronica sempre più ampio, investì grossi capitali nella ricerca e nello sviluppo di altri prodotti, quali orologi digitali, televisori tascabili e misuratori elettronici. Nel 1975 fu lanciato un *multimeter* digitale a basso costo, il DM2, contempora-

neamente a un orologio da polso, il *Black Watch*.

Fino al 1976 la *Sinclair Radionics* ha goduto di 15 anni di crescita dei profitti, grazie ai grossi giri di affari; in questo periodo la società sostenne piccole perdite, dovute alla difficoltà di progettazione del *Black Watch*.

Di conseguenza vi fu una carenza di fondi disponibili per l'ultimo stadio del progetto *pocket-tv*, ma i fondi necessari furono trovati egualmente; nel dicembre del '76 la *National Enterprise Board* decise di fare un'aggiunta di capitale e fu così che nel gennaio del '77 fu lanciato il *pocket-tv* di soli 2 pollici, primo al mondo per le sue dimensioni, dopo un programma di studi, ricerche e sviluppo durato 12 anni, e che era costato complessivamente vari miliardi di lire.

La ricerca sulla miniaturizzazione, scopo principale di Mr. Sinclair, fece progressi così notevoli che nel novembre del '78 fu introdotta una nuova versione di TV tascabile, il *microvision*, molto più economico del suo predecessore.

Altri settori dell'elettronica di consumo furono invasi dalla *Sinclair Radionics* tra il '77 ed il '78: prima una calcolatrice tascabile, caratterizzata da importanti innovazioni tecnologiche, poi due nuovi tipi di calcolatrici di ufficio, ed infine la *Cambridge Programmable*, una calcolatrice che costava il 75% in meno delle concorrenti.

Nel settembre 1978 la società decise di rafforzare la produzione delle calcolatrici tascabili, concentrandosi su un modello il cui costo era inferiore alle centomila lire.

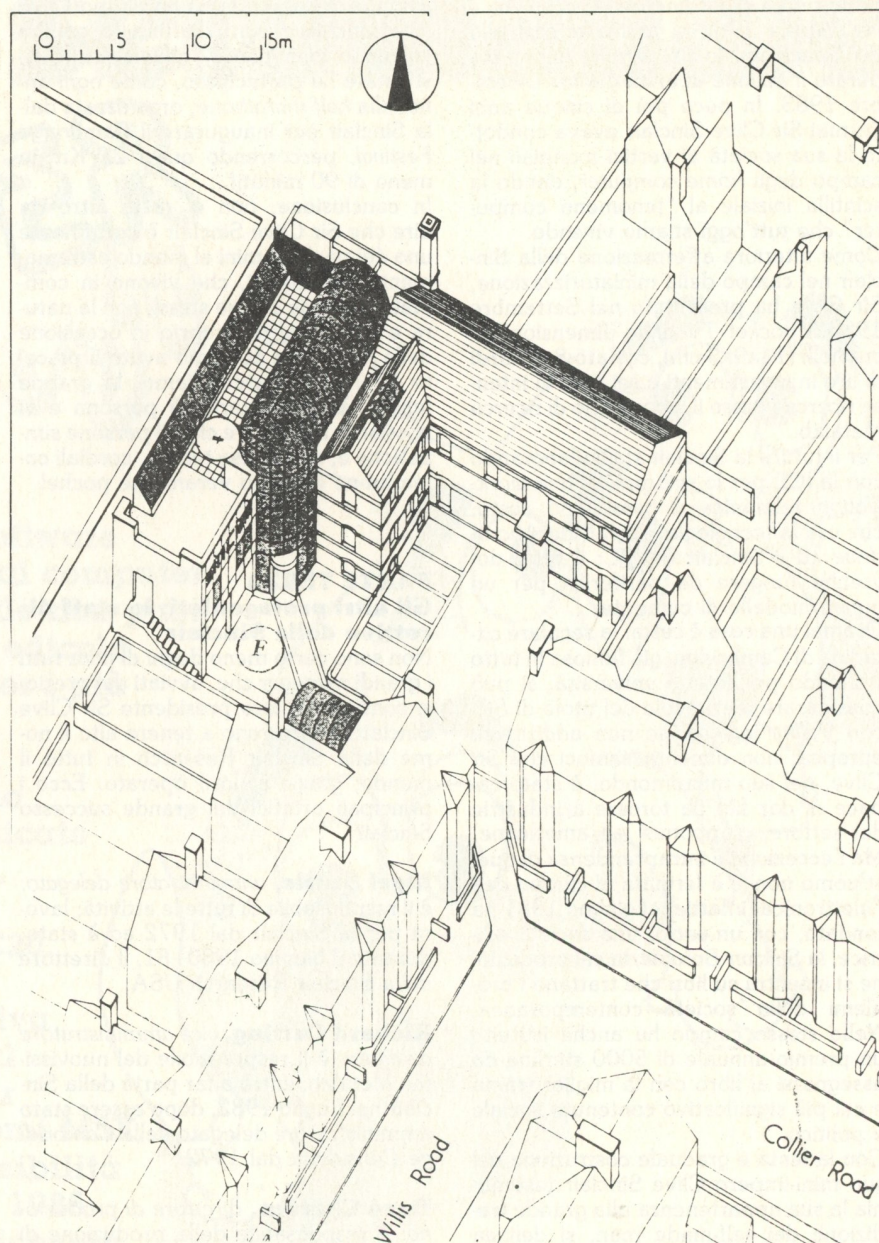
Nel campo dei misuratori elettronici Sinclair introdusse sul mercato tre nuovi modelli di multimetri digitali, che condussero la società ai primi posti nella produzione mondiale di elettronica di consumo.

Si arriva così al 1979, anno in cui la *Sinclair* e la N.E.B. decisero di scindersi, poiché i loro interessi non erano più coincidenti: la N.E.B. si dedicò completamente al campo dei misuratori elettronici, mentre Clive Sinclair, dopo aver sciolto la *Sinclair Radionics*, fondò nel luglio 1979 la ben nota ed ormai famosissima *Sinclair Research*, per lanciarsi esclusivamente nel campo dell'elettronica d'avanguardia.

SECONDA PARTE Il trionfo di Clive Sinclair: 1979-1984

La *Sinclair Research* lanciò il suo primo prodotto nel Febbraio 1980: il vecchio e caro ZX80, capostipite degli *home computer* Sinclair; esso fu venduto in più di trenta paesi ad un prezzo di poche centinaia di migliaia di lire.

Forse non tutti i *sinclairisti* lo conoscono, e perciò ne vogliamo ricordare qualche caratteristica, dato che noi stessi vi abbiamo fatto esperienza! Poco più piccolo del foglio su cui state leggendo (!), esso era costituito esternamente da una scatola di plastica bianca e da una ta-



F - Foto

stiera piatta a membrana. Nel suo interno vi erano 14 circuiti integrati, compresa una Z80 (e non Z80A), una ROM da 8K e una RAM di... 1K(!!!).

Oggi uno ZX80 se lo potrebbe comprare solo un antiquario, un amatore di pezzi da museo, ma a quell'epoca (il ricordo ci sembra lontano, poiché anche pochi anni di evoluzione elettronica si potrebbero paragonare ad un secolo di storia!) Sir Clive predisse che il primo vero *home computer* del mondo avrebbe avuto un successo strepitoso; e così fu: prima che la produzione fosse bloccata nell'Agosto 1981 furono venduti più di 100.000 ZX80, di cui il 60% riguardava il solo mercato estero. Secon-

do le previsioni di Mr. Sinclair la grossa diffusione del computer "domestico" era ormai avviata, con sviluppi futuri che solo cinque anni fa non si sarebbero certo immaginati! Clive Sinclair aveva così diffuso in tutto il mondo l'idea dell'*home computer*.

Allo ZX80 successe in breve tempo lo ZX81, versione riveduta e corretta, che sintetizzava i precedenti 14 integrati in soli 4 (CPU,ROM, RAM,ULA); il suo costo è oggi sceso sotto le 100.000 lire (quasi una calcolatrice!), dopo aver vinto nel 1982 il premio del Consiglio di Progetto e dopo essere stato venduto in oltre 50 paesi, totalizzando una vendita complessiva di circa 1.150.000

pezzi prima della fine del 1983.

Nell'Aprile 1982 si arrivava così allo ZX Spectrum, le cui vendite hanno superato il milione di unità già nel Dicembre 1983. In poco più di cinque anni l'ormai Sir Clive Sinclair aveva condotto la sua società ai vertici mondiali nel campo degli *home computer*, dando la scintilla iniziale al "fenomeno computer" che tutt'oggi stiamo vivendo.

Come ulteriore affermazione della Sinclair nel campo della miniaturizzazione, Sir Clive ha presentato nel Settembre 1983 il *pocket TV* dalle dimensioni incredibili (14x9x3 cm), costato 8 miliardi di lire in investimenti e sei anni di intensa ricerca presso il laboratorio di ricerca Metalab.

Per il futuro la Sinclair sta collaborando con la ICL per lo sviluppo di una *workstation* telefonica a bassissimo costo, che usa la tecnologia Sinclair dei *Micro-drive* ed il Sinclair *BASIC*; il tutto dovrebbe fungere da periferica per un nuovo modello di computer.

Oramai una cosa è certa: la secolare cittadina di Cambridge, già famosa in tutto il mondo per le sue università, si può considerare meritevole del titolo di *Silicon Valley* inglese, se non addirittura europea; non dimentichiamoci che Sir Clive, nel suo micromondo, è stato capace di dar filo da torcere a industrie del settore, giapponesi ed americane. Ma l'eccezionale intraprendenza di quest'uomo non si è fermata al campo dell'elettronica; infatti, nel giugno 1981 ha fondato, con un socio, una società editrice, la *Sinclair-Browne*, la cui produzione si incentra su libri che trattano i problemi della società contemporanea. Nello stesso campo ha anche istituito un premio annuale di 5000 sterline da assegnarsi al libro con la miglior trama e col più significativo contenuto sociale e politico.

Con la lenta e graduale costruzione del suo mini-impero Clive Sinclair testimonia la sua appartenenza alla grande tradizione dei *self-made man*, si delinea così sempre più la figura di un uomo dalle grandi capacità, le cui azioni sono guidate da saggia riflessione nonché soluzioni geniali, la cui applicazione non può altro che generare successo e ricchezza; non a caso egli, oltre ad essere il presidente della Sinclair Research e del Mensa - associazione mondiale di super-intelligenze - è stato nominato, nel 1983, Young Business Man ed esponente della *Computer Age* della decade.

Infine, nel giugno 1983 ha ricevuto il maggior riconoscimento ufficiale: è stato nominato baronetto, in quanto presente nella lista d'onore della regina.

Oggi Sir Clive divide il suo tempo tra Cambridge e Londra quale grande uomo d'affari; oltre all'elettronica i suoi interessi comprendono il teatro, la poesia, la musica (è il curatore dell'orchestra sinfonica di Cambridge), le scienze matematiche ed infine la corsa; e proprio a proposito di corsa vi possiamo

assicurare che il nostro carissimo Clive si sgranchisce ogni mattina le gambe correndo per ben 8 chilometri; quest'estate ha partecipato, come ogni anno, alla *half-marathone*, organizzata dalla Sinclair per inaugurare il *Cambridge Festival*, percorrendo quasi 22 Km in meno di 90 minuti!...

In conclusione, non ci resta altro da dire che Sir Clive Sinclair è certamente uno dei pochi uomini al mondo estremamente polivalenti, che vivono in completa armonia con se stessi, con la natura e con gli altri: proprio in occasione della maratona abbiamo avuto il piacere (e, non vi nascondiamo, la grande felicità) di conoscerlo di persona e vi possiamo assicurare che di persone simpatiche e, allo stesso tempo, cordiali come lui ne esistono veramente poche!

PARTE TERZA

Gli altri protagonisti: lo staff direttivo della Sinclair

Non sono certo meno degni di nota tutti i grandi *manager* che, avviati dai preziosi consigli del loro presidente Sir Clive Sinclair, concorrono a tenere alto il nome della Sinclair Research in tutto il mondo, grazie al loro operato. Ecco i principali artefici del grande successo Sinclair:

Nigel Searle, amministratore delegato: è il responsabile di tutte le attività; lavora per la Sinclair dal 1972 ed è stato, durante il biennio 1980-81, il direttore della Sinclair Research USA.

Richard Cutting, vice amministratore delegato: è il responsabile del nuovissimo Metalab; entrò a far parte della Sinclair nel Luglio 1982, dopo essere stato amministratore delegato della *Cambridge Consultant* dal 1972.

Dave Chatten, direttore di produzione: è responsabile della produzione di tutti gli articoli Sinclair; è entrato alla Sinclair nel Giugno 1970 e dal 1980 lavora alla progettazione computer.

Bill Matthews, segretario della Società e direttore dei Servizi Corporativi: entrò alla Sinclair nel 1982, dopo esser stato vice presidente esecutivo di una società di Houston.

David Southward, direttore della ricerca e dello sviluppo progettazione: si imbattè nella Sinclair nel 1977 ed è stato unico responsabile della pianificazione e quindi dello sviluppo del progetto del *pocket TV*.

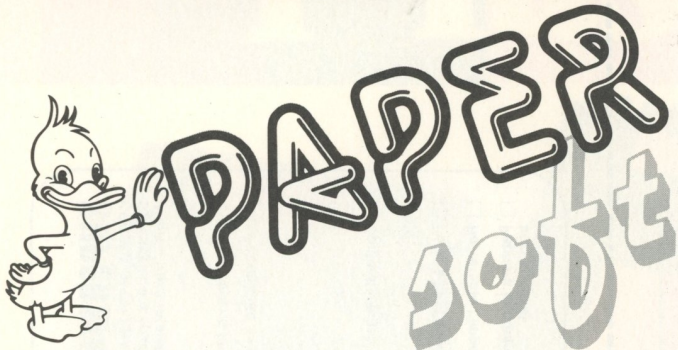
Jim Westwood, vice direttore della ricerca: è il più vecchio collaboratore di Clive Sinclair; infatti ha partecipato alla realizzazione dei principali progetti sin dal lontano 1963. È stato specialmente coinvolto nella progettazione e nello sviluppo dello ZX80.

ERRATA CORRIGE

A causa di un disguido la paternità del programma "Boxe" pubblicato su SUPERSINC datato Luglio/Agosto 1985 è stata erroneamente attribuita a Gianpaolo Gentili.

In realtà il programma è stato realizzato da Pietro Rosato di Pescara. Ci scusiamo per l'inconveniente con gli interessati.

Il tuo settimanale



Il primo settimanale di software su carta per il tuo personal computer. Un appuntamento al quale non puoi mancare ogni venerdì in edicola. Nato in edizione unica, per i più diffusi home e personal computer, dal 1 Aprile, allo stesso prezzo, saranno in edicola le versioni dedicate ai prodotti Sinclair (ZX81, ZX Spectrum e QL) e ai prodotti Commodore (VIC20, C16 e C64).

In più un'edizione esclusiva per gli utenti dei personal Apple, gli home computer Texas e i nuovissimi sistemi MSX. Una grossa novità: la possibilità di abbonarsi dal 1 Aprile 1985 al 31 Dicembre 1985 per ricevere 38 numeri del settimanale comodamente a casa con un notevole risparmio. Chi si abbona riceverà ogni 15 giorni 2 numeri di PAPERSoft

in anticipo rispetto all'uscita in edicola. **Non perdere questa occasione!**

Compila subito il coupon qui riportato e spediscilo a:

**J.soft
V.le Restelli, 5
20124 MILANO**



EDITRICE **J.soft**

Tagliando abbonamento a PaperSoft da inviare in busta chiusa a:
J.soft - V.le Restelli, 5 - 20124 MILANO

- Abbonamento a 38 numeri di PaperSoft al prezzo speciale di L. 30.000 anziché L. 38.000
- edizione Sinclair (PS01) edizione Commodore (PC01) edizione Apple/TI99/MSX (PV01)
- contanti allegati assegno allegato n. _____
- ho versato l'importo sul c.c.p. n. 19445204 intestato a J.soft - MILANO

Cognome _____ Nome _____

Via _____ Città _____

CAP _____ Prov. _____ Data _____

Firma _____

SUPERSING

CEOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.sofi - Viale Restelli, 5 - 20124 Milano - Tel. 6888228

LIBRI

Imparate a programmare
con lo ZX Spectrum
Giochi con il computer - come
funzionano come si vince
Primi passi in BASIC - una facile guida
per scrivere i programmi

cod. AJAK037 L. 25.600
cod. AJAC061 L. 8.100
cod. AJAK040 L. 8.100

Ordine i seguenti libri per un importo totale di L. + L. 2.000
come contributo fisso per spese di spedizione

Cod. Cod.

Cod. Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp n° 19445204 intestato a J.sofi - Milano

Pagherò in contassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

Nome
Cognome

Via
Città C.A.P.

Se richiesta fattura:
Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSING INPUT/OUTPUT

La rubrica INPUT/OUTPUT è gratuita ed aperta a
tutti i lettori. Chi desidera comprare, vendere o cam-
biare hardware o software deve inviare il tagliando
a J.sofi - Viale Restelli, 5 - 20124 MILANO

- COMPRO VENDO CAMBIO
- ZX80 ZX81 ZX Spectrum QL SOFTWARE
- PERIF.

Nome
Via
Città

Cognome
C.A.P.
Tel.

SUPERSING è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

Nome
Via
Città

Cognome
C.A.P.
Tel.

i Tascabili



Jackson, naturalmente.

I tascabili Jackson sono uno strumento prezioso per chi lavora con il computer.

SINCLAIR SPECTRUM cod. 017H
VIC 20 cod. 005H
COMMODORE 64 cod. 002H
PC IBM cod. 018H
APPLE IIc cod. 003H
SHARP MZ80A cod. 014H
LA PROGRAMMAZIONE cod. 004H
WORD STAR cod. 008H
UNIX cod. 009H
LOGO cod. 020H
MS-DOS cod. 019H
PROGRAMMI DI STATISTICA cod. 015H
CP/M cod. 011H
PC-DOS cod. 012H

BASIC cod. 007H
ASSEMBLER Z80 cod. 016H
ASSEMBLER 6502 cod. 013H
COBOL cod. 001H
FORTRAN 77 cod. 010H
PASCAL cod. 006H



OGNI TASCABILE COSTA L. 8.500

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Titolo	Prezzo unitario	Prezzo totale
Totale				

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esecuzione del contributo spese di spedizione:

Allego assegno della Banca Allego fotocopia del versamento sul c/c n. 11666203 a voi intestato

N° _____ Allego fotocopia di versamento su vaglia postale e voi intestato

Nome e Cognome _____

Via _____

Cap _____

Città _____

Prov. _____

Data _____

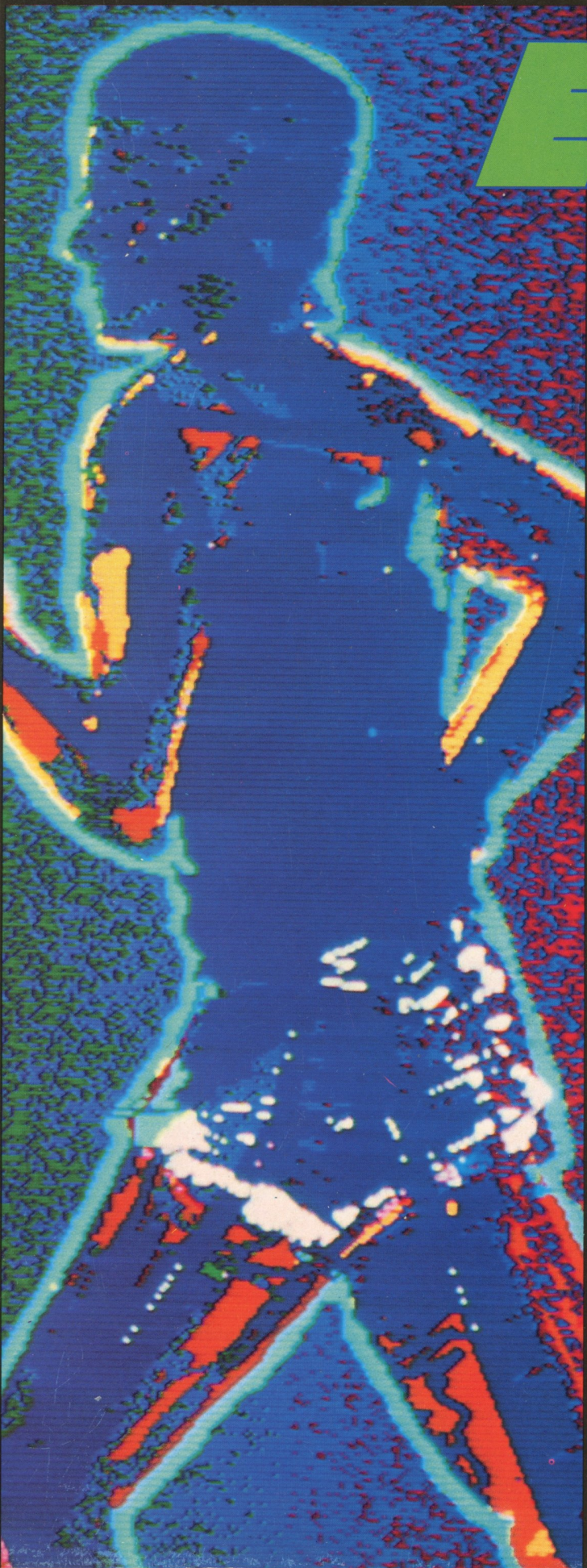
Firma _____

Spazio riservato alle Aziende Si richiede l'emissione di fattura

ORDINE
MINIMO
L. 50.000

Partita I.V.A. _____

NON FARTI SUPERARE DAL PROGRESSO

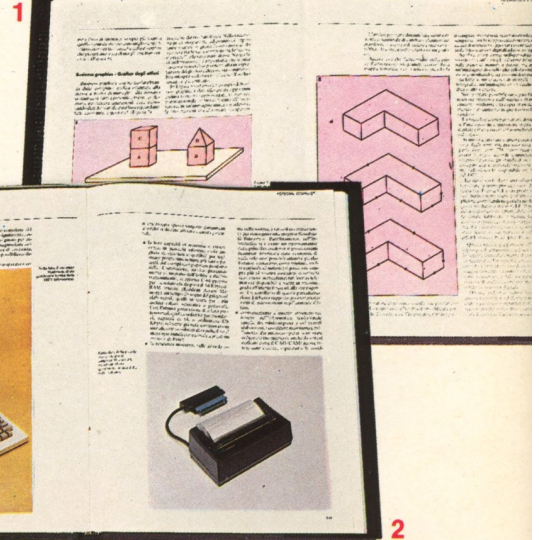


Aggiornati con gli "Aggiornamenti"

ENCICLOPEDIA DI ELETTRONICA & INFORMATICA

nuovidea

20 FASCICOLI SETTIMANALI DA RILEGARE
IN DUE NUOVI E SPLENDIDI VOLUMI



PER TE, PER IL TUO LAVORO, PER I TUOI STUDI...

1 **Aggiornamenti**

le nuove conquiste dell'Elettronica di Base, delle Comunicazioni, dell'Elettronica Digitale, dei Microprocessori, dell'Informatica...

2 **Il personal computer**

tutto quello che c'è da sapere sul Personal Computer: che cos'è e cosa fa; come fa e come si fa; i linguaggi di programmazione; le applicazioni...

E.I. si aggiorna e ti aggiorna

il 18 settembre

batti sul tempo il progresso!
Corri in edicola a comprare il primo fascicolo.

Solo L. 2500



**ANCORA UNA VOLTA
"PRIMI SUL FUTURO"**



**GRUPPO
EDITORIALE
JACKSON**

SAN FRANCISCO - LONDRA - MILANO